

Ortung und Navigation mit mobilen Geräten in Gebäuden

Abschlussarbeit

zur Erlangung des akademischen Grades
Master of Science (M. Sc.)

an der
Hochschule für Technik und Wirtschaft Berlin
Fachbereich Wirtschaftswissenschaften II
Studiengang Angewandte Informatik

1. Prüfer: Prof. Dr. Jürgen Sieck
2. Prüfer: Matthias Prellwitz, M. Sc.

eingereicht von: Andreas Bilke, B. Sc.
Matrikelnummer: 530318

Datum: 27. Februar 2012

Danksagung

Zunächst möchte ich meinen Betreuern Prof. Dr. Jürgen Sieck und Matthias Prellwitz für ihre Unterstützung und konstruktiven Anmerkungen danken.

Des Weiteren danke ich allen, die meine Masterarbeit gelesen und durch ihre Kritik die Arbeit qualitativ verbessert haben. In der folgenden Liste sind, alphabetisch sortiert, die Unterstützer aufgeführt:

- Andreas
- Björn
- Elly
- Stephan

Ganz besonderer Dank geht an meine zukünftige Ehefrau Anita. Ihr habe ich es unter anderem zu verdanken, dass ich während dieser Zeit von anderen Pflichten befreit wurde und somit meine gesamte Aufmerksamkeit dieser Arbeit widmen konnte.

Abstract

Many people have difficulties getting their bearings when entering an unknown building. However this problem can be solved by locating and navigation on a mobile phone.

This thesis presents a locating and navigation system which bases on recognising geomagnetic field disturbances and ambient light. A particle filter is applied on the locating problem. It is used to fuse the data of both sensors and track the mobile phone.

The prototypical implementation of locating and navigation takes place on an Android tablet. The tests concerning the locating's accuracy occurred in an office building. In the course of these tests an average locating error of 4 m was achieved.

Kurzbeschreibung

Beim Betreten eines unbekanntes Gebäudes stellt sich für den Besucher das Problem, sich in selbigem zu orientieren und zum gewünschten Ziel zu gelangen. Über eine Ortung und Navigation auf einem mobilen Gerät kann dieses jedoch gelöst werden.

In dieser Arbeit wird ein Lokalisierungs- und Navigationssystem vorgestellt, welches auf dem Wiedererkennen von Störungen des Erdmagnetfeldes und Umgebungslicht basiert. Zur Fusion beider Sensordaten und dem Verfolgen des mobilen Gerätes wird ein Partikel-Filter auf das Ortungsproblem angewandt.

Die prototypische Implementierung der Lokalisierung und Navigation erfolgt auf einem Android Tablet. Die Tests zur Ortungsgenauigkeit fanden in einem Bürogebäude statt. Dabei wurde ein durchschnittlicher Ortungsfehler von 4 m erreicht.

Inhaltsverzeichnis

1. Einleitung	1
1.1. Motivation und Problembeschreibung	1
1.2. Ziel der Arbeit	2
1.3. Aufbau der Arbeit	2
2. Grundlagen und Stand der Technik	4
2.1. Verfahren zur Ortung	4
2.1.1. TDoA/ToA	5
2.1.2. RSSI	6
2.1.3. AoA	7
2.2. Fingerprintverfahren	7
2.2.1. Euklidische Distanz	8
2.2.2. Kosinus-Ähnlichkeit	8
2.2.3. Wahrscheinlichkeitsbasierter Vergleich	9
2.3. Technologien zur Ortung	10
2.3.1. W-LAN	11
2.3.2. RFID	12
2.3.3. Magnetfelder	12
2.3.4. Visuelle Marker	13
2.3.5. Weitere Ansätze	13
2.3.6. Zusammenfassung	14
2.4. Wahrscheinlichkeitsbasierte Positionsschätzung	14
2.4.1. Wahrscheinlichkeitstheoretische Grundlagen	15
2.4.2. Markov Kette	17
2.4.3. Rekursive Bayes-Filter	18
2.4.4. Kalman-Filter	19
2.5. Partikel-Filter	19
2.5.1. Grundidee	19
2.5.2. Darstellung der Dichtefunktion	20
2.5.3. Allgemeiner Partikel-Filter	21
2.5.4. Bewegungsmodell	22
2.5.5. Messmodell	23
2.5.6. Verhindern von Partikeldegeneration durch Resampling	23
2.5.7. Ermitteln des Systemzustands	25

2.6. Navigation in Gebäuden	25
2.6.1. Stand der Technik	25
2.6.2. Umsetzung der Navigation	27
3. Entwicklung einer Ortungsstrategie	29
3.1. Auswahl der verwendeten Sensorik	29
3.1.1. Versuchsaufbau	30
3.1.2. Ortung durch Wiedererkennen von Magnetfeldern	31
3.1.3. Ortung über spezifische Lichtverhältnisse	35
3.1.4. Luftdruck zur Ermittlung der Etage	35
3.2. Partikel-Filter zur Zusammenführung von Sensordaten	36
3.2.1. Auswahl des Filter-Algorithmus	36
3.2.2. Zustandsraum	37
3.2.3. Genutzter Messvektor	37
3.2.4. Repräsentation einer Karte	38
3.2.5. Bewegungsmodell	38
3.2.6. Messmodell	41
3.3. Ablauf einer Ortung	43
3.3.1. Offline-Phase	44
3.3.2. Online-Phase	44
4. Navigation in Gebäuden	46
4.1. Kartenrepräsentation	46
4.1.1. Nutzung der Fingerprints	46
4.1.2. Semantischer Zusammenhang der Daten	47
4.1.3. Kostenfunktion zwischen den Knoten	47
4.2. Ablauf der Navigation	49
4.2.1. Beginn des Routings	49
4.2.2. Pfaderstellung	49
4.2.3. Periodische Positionsbestimmung	49
4.2.4. Navigationsmodul	50
4.3. Ausgabe von Navigationsanweisungen	51
5. Prototypische Umsetzung	53
5.1. Entwicklungsumgebung	53
5.1.1. Softwareplattform	53
5.1.2. Endgerät	54
5.2. Softwaredesign	54
5.2.1. Anforderungen	54
5.2.2. Systemstruktur	55
5.3. Aufnehmen der Fingerprints	55
5.4. Lokalisationsmodul	57
5.4.1. Partikel-Filter Initialisierung	58

5.4.2.	Ausführen eines Filterschritts	59
5.4.3.	Ermittlung des Systemzustands	61
5.4.4.	Zuordnung zwischen Partikelposition und der Visualisierung	62
5.5.	Navigationsmodul	62
6.	Evaluation der Ortungs- und Navigationslösung	64
6.1.	Methodik	64
6.1.1.	Aufzeichnen der Messdaten	64
6.1.2.	Box-Plots zum Vergleich der Konfigurationen	65
6.2.	Genauigkeit bei der Ortung	66
6.2.1.	Wände zur Verbesserung der Ortung	66
6.2.2.	Himmelsrichtung zum Ausgleich des Fehlers bei der Vektornormierung . . .	67
6.2.3.	Vergleich von Vektorähnlichkeiten	67
6.2.4.	Nutzen von mehreren Sensoren	68
6.2.5.	Einfluss der Partikelanzahl	69
6.2.6.	Fehler bei den unterschiedlichen Messungen	71
6.2.7.	Auswahl der besten Algorithmenkonfiguration	71
6.3.	Fehlerbetrachtung	72
6.4.	Vergleich zu anderen Ortungslösungen	74
6.5.	Evaluation der Navigation	75
6.6.	Zusammenfassung	75
7.	Schlussbetrachtung	77
7.1.	Geschaffene Lösung	77
7.2.	Weitere Aktivitäten	78
	Literaturverzeichnis	I
	Abbildungsverzeichnis	V
	Tabellenverzeichnis	VI
	Algorithmenverzeichnis	VII
	Abkürzungsverzeichnis	VIII
	A. Inhalt der CD-ROM	IX

1. Einleitung

Beim Betreten eines unbekanntes Gebäudes stellt sich für den Besucher das Problem, sich in selbigem zu orientieren und zum gewünschten Ziel zu gelangen. Möchte eine Person an einen bestimmten Punkt gelangen, hat sie zum Beispiel die Möglichkeit einen statischen Gebäudeplan zu nutzen. Auf dem Weg zum Ziel muss sie sich allerdings die Route merken und die Übersichtsansicht auf die reale Situation übertragen.

Dieses allgemeine Problem besteht ebenfalls, wenn Personen beispielsweise große Ämter aufsuchen und zu einem bestimmten Büro gelangen wollen oder alltäglich in großen Einkaufszentren auf der gezielten Suche nach einem Geschäft sind. In diesen Situationen wäre es von Vorteil, wenn mit Hilfe eines mobilen Geräts eine Route zum Zielpunkt berechnet und diese in Echtzeit dem Nutzer zur Verfügung gestellt werden kann.

1.1. Motivation und Problembeschreibung

Diese Probleme können durch eine Ortung und die darauf aufbauende Navigation gelöst werden. Die wachsende Anzahl an Mobiltelefonen¹ ermöglicht es ohne großen technischen Aufwand für den Benutzer solche Funktionen bereitzustellen. Neben dem oben genannten Szenario wird Ortung in Gebäuden sowie in der Logistik zur Güterverfolgung gebraucht. Dies zeigt die starke Bedeutung dieses Themenkomplexes.

Ziel bei der Ortung ist es aus einem technischen System Standortdaten, meist in Form von zwei beziehungsweise dreidimensionalen Koordinaten, zu gewinnen. Satellitengestützte Systeme, wie Global Positioning System (GPS), eignen sich aufgrund der Signaldämpfung nur bedingt für die Ortung innerhalb von Gebäuden. Es wurden bisher viele alternative Systeme aufgrund dessen entwickelt.

Die meisten der existierenden Systeme haben aber den Nachteil, dass sie entweder ungenau sind oder einen erheblichen Installations- und Wartungsaufwand besitzen.

¹vgl. http://www.bitkom.org/files/documents/BITKOM-Praesentation_PK_Mobile_World_15_02_2012.pdf, Abruf 22. Februar 2012

1.2. Ziel der Arbeit

In dieser Arbeit soll ein System geschaffen werden, um innerhalb von Gebäuden eine Ortung und darauf aufbauend eine Navigation durchzuführen. Der Fokus dieser Arbeit liegt dabei auf der Ortungslösung.

Es soll daher untersucht werden, mit welchen Möglichkeiten mobile Geräte in Gebäuden geortet werden können. Dazu werden verschiedene Lokalisierungsverfahren auf ihre Tauglichkeit überprüft und eigene Lösungsansätze entwickelt. Die Ortung erfolgt über das Wiedererkennen des Erdmagnetfeldes und von eindeutigen Lichtverhältnissen. Diese zwei Sensordaten müssen zusammengeführt werden, um daraus eine neue Position zu generieren.

Die entwickelte Ortungslösung soll hinsichtlich ihrer Positionsgenauigkeit und dem Installationsaufwand mit anderen Verfahren verglichen werden. Mittels Navigation soll es zum Beispiel möglich sein eine Route durch ein Gebäude, beispielsweise vom Gebäudeeingang zu einem bestimmten Büro, zu finden. Das zu erstellende System soll auf einem mobilen Endgerät wie einem Tabletcomputer lauffähig sein. Es ist daher notwendig, dass die genutzten Algorithmen effizient sind.

1.3. Aufbau der Arbeit

Im Grundlagenteil wird sich mit dem derzeitige Stand der Technik für die Ortung innerhalb von Gebäuden auseinander gesetzt. Die bisher existierenden Systeme werden beschrieben und ihre Vor- und Nachteile gegenüber gestellt. Weiterhin wird in diesem Abschnitt der für die Sensorfusion eingesetzte Partikel-Filter (PF) beschrieben. Am Ende des Abschnitts wird ein Algorithmus zur Berechnung des kürzesten Pfades innerhalb eines Graphen vorgestellt.

Im darauf folgenden Kapitel wird eine Ortungsstrategie entworfen. Dazu wird zunächst untersucht, ob die Nutzung von Magnetfeldern und Licht überhaupt geeignet ist, um eine Lokalisierung durchzuführen. Weiterhin wird in diesem Abschnitt eine Erweiterung der zweidimensionalen Ortung eingeführt, indem weitere Sensordaten, die einem mobilen Gerät zur Verfügung stehen, ausgewertet werden. Der im Grundlagenteil allgemein beschriebene PF wird in diesem Abschnitt auf das Problem der Lokalisierung angewandt und beschrieben. Die Ortung erfolgt über die Nutzung von Fingerprints für die genannten Sensoren.

Die Navigationsstrategie beschäftigt sich mit der Umsetzung der Navigation auf dem mobilen Gerät. Dazu wird beschrieben wie aus einer Umgebungskarte und den Positionen der Fingerprints ein Graph erstellt werden kann. Dies bietet den Vorteil, dass für ein Gebäude kein separater Graph angelegt werden muss, da dieser aus den bereits genutzten Daten extrahiert werden kann.

Innerhalb der prototypischen Umsetzung wird die Implementierung des Ortungs- und Navigationsalgorithmus auf einem Tabletcomputer beschrieben.

Der PF ist zunächst ein allgemeines Konzept, welches auf ein spezifisches Problem angewandt werden muss. Dieser Filter ist demnach frei erweiterbar. Die in der Literatur vorgestellten Erweiterungen des Filters werden im Abschnitt zur Evaluation hinsichtlich ihres Einflusses auf die Ortung untersucht. Dabei wird eine Konfiguration ausgewählt, welche für das in dieser Arbeit genutzte Szenario die besten Ergebnisse liefert. Diese werden im Anschluss mit denen von anderen Arbeiten verglichen und bewertet. Zusätzlich wird in jenem Abschnitt die Funktionsfähigkeit des Navigationsalgorithmus gezeigt.

Der Schluss dieser Arbeit bildet eine Zusammenfassung zur geschaffenen Lösung. Zusätzlich wird beschrieben, wie die Ergebnisse dieser Arbeit in späteren Projekten eingesetzt und wie die Lösung verbessert werden könnte.

2. Grundlagen und Stand der Technik

Diese Arbeit umfasst die Ortung und Navigation in Gebäuden. Es gilt somit zu klären, welche Teilprobleme zu lösen sind. Eine Navigation setzt voraus, dass geortet werden kann. Ein wesentlicher Bestandteil dieser Arbeit ist demnach eine Ortungsstrategie zu entwickeln, welche in Gebäuden adäquat funktioniert. Aufbauend auf einer Position kann eine Navigationslösung entwickelt werden.

Zuerst wird der aktuelle Stand der Technik und damit einhergehend die Vor- und Nachteile der jeweiligen Techniken beschrieben. Aus dem aktuellen Stand der Forschung werden Lösungsansätze ausgewählt und näher erläutert.

Weiterhin werden die theoretischen Grundlagen beschrieben um, wenn die Distanz von einem bekannten Knoten zum mobilen Gerät bekannt ist, daraus die Position des mobilen Gerätes zu berechnen. Zur Ermittlung dieser Distanz, stehen verschiedene Technologien wie Wireless Local Area Network (W-LAN) oder Radio Frequency Identification (RFID) zur Verfügung.

Neben diesen Vorgehensweisen existieren auch statistische Varianten zur Positionsermittlung. Diese so genannten Fingerprints werden vor der eigentlichen Ortung aufgenommen und zusammen mit ihrer Position gespeichert. Beim späteren Orten werden erneut Messwerte ermittelt und diese mit den Fingerprints auf Ähnlichkeit überprüft.

Die Genauigkeit von Ortungssystemen kann verbessert werden, indem neben den aktuellen Messwerten ebenso die vergangenen Positionen einbezogen werden. Der in dieser Arbeit genutzte PF wird in diesem Kapitel beschrieben.

Der letzte Abschnitt in diesem Kapitel beschäftigt sich mit aktuellen Forschungen zur Navigation in Gebäuden sowie den dort genutzten Algorithmen.

2.1. Verfahren zur Ortung

Die Ortungstechniken in diesem Abschnitt haben gemein, dass für die Funktionsweise statische Knoten und ihre Position vor der Ortung bekannt sein müssen. Der mobile Knoten, welcher geortet werden soll, muss mittels Sensorik eine Relation zum statischen Knoten herstellen. Dies kann zum einen über Distanzmessungen zwischen mobilem Gerät und mehreren festen Stationen, aber genauso durch Messung des Signaleinfallswinkels zwischen Sendestation und mobilem Gerät erreicht werden.

2.1.1. TDoA/ToA

Die Ermittlung der Distanz beim Time Difference of Arrival (TDoA) und des Time of Arrival (ToA) Verfahren basieren auf dem Messen von Signallaufzeiten zwischen einem mobilen Knoten und einer fest installierten Station [Str+08, S. 44].

Beim ToA Verfahren werden die absoluten Zeitdifferenzen gemessen und daraus die Entfernung zwischen Sender und Empfänger bestimmt. Dazu schickt der Sender zu einem Zeitpunkt t_0 ein Signal, der Empfänger erhält dies zu einem Zeitpunkt t_1 . Aus der Differenz beider Zeiten und der Ausbreitungsgeschwindigkeit des Signals kann anschließend mittels

$$d = (t_1 - t_0) \cdot c \tag{2.1}$$

die Strecke errechnet werden. Wird zur Laufzeitmessung ein Funksignal genutzt, ist die Ausbreitungsgeschwindigkeit gleich der Lichtgeschwindigkeit c . Die Sendezeit t_0 wurde in dem Funksignal kodiert. Für diese Berechnung ist es notwendig, dass beide Kommunikationspartner synchronisierte Uhren besitzen. Dies kann zum einen durch den Einsatz von Atomuhren, wie sie bei satellitengestützter Ortung (GPS) genutzt werden, oder zum anderen durch Uhrensynchronisationsprotokolle wie dem Network- oder Precision Time Protocol geschehen. Eine unzureichend genaue Uhr hat demnach Auswirkungen auf die Präzision des Systems [Röh08, S. 85].

Aus der Distanz d zu drei Stationen kann, wie in Abbildung 2.1 visualisiert, der Standort des mobilen Gerätes bestimmt werden.

In der zu erstellenden Ortungslösung ist es jedoch nicht praktikabel die Uhren zwischen mobilem Gerät und einer Station abzugleichen. Im Gegensatz dazu benötigt das TDoA Verfahren keine synchronisierte Uhr zwischen dem mobilen Gerät und der Station, aber immer noch eine zwischen allen Basisstationen. Dabei sendet ein mobiles Gerät ein Signal aus, welches von mindestens drei Stationen empfangen werden muss. Wenn diese Stationen eine unterschiedliche Distanz zum Gerät besitzen, werden die Signale zu unterschiedlichen Zeiten empfangen. Die Zeitdifferenz lässt sich über die Ausbreitungsgeschwindigkeit auf eine Distanzdifferenz zurückführen [Str+08, S. 41 ff.].

Zur Verdeutlichung wird der Fall mit zwei Empfangsstationen und einem mobilen Gerät betrachtet. Empfängt die Station M2 das Signal später als M1, dann befindet sich das mobile Gerät weiter von M2 weg als von M1. Aus dieser Relation entstehen eine Menge von möglichen Aufenthaltsorten des mobilen Gerätes, welche mathematisch im zweidimensionalen über eine Hyperbel beschrieben werden können. Durch die Nutzung einer dritten Distanzdifferenz wird eine zweite Hyperbel aufgespannt; die Position des mobilen Gerätes ist dann der Schnittpunkt beider Funktionen [Str+08, S. 41 ff.].

Beide Varianten haben die Voraussetzung, dass entweder Sender und Empfänger beziehungsweise mehrere Empfänger untereinander synchronisierte Uhren besitzen müssen. Dies ist in der Regel

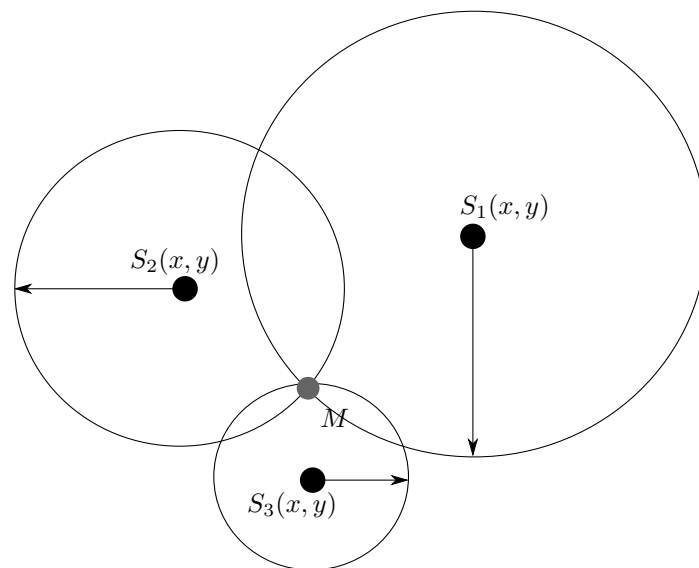


Abbildung 2.1.: Ermitteln der Position im zweidimensionalen Raum eines mobilen Knotens aus der Distanz zu drei festen Stationen

mit aktuellen mobilen Geräten, wie Mobiltelefonen, nicht umsetzbar. Dadurch eignen sich diese Methoden nicht für die in dieser Arbeit angestrebte Ortung [Röh08, S. 85].

2.1.2. RSSI

Beim Received Signal Strength Indicator (RSSI) Verfahren wird die Distanz zwischen mobilem Gerät und Station durch die Signaldämpfung ermittelt. Die Empfangsstärke sinkt mit steigendem Abstand zur Station. Diese Dämpfung kann durch ein so genanntes *Propagation Model* vorhergesagt werden [Str+08, S. 42].

Aus dem Standort von mindestens drei Stationen und der Distanz des mobilen Knotens zu ihnen, kann die Position über Trilateration berechnet werden. In Abbildung 2.1 ist dieses Verfahren grafisch veranschaulicht. Ist die Distanz zu einer Station bekannt, kann nur die Aussage getroffen werden, dass sich das mobile Gerät auf einem Kreis um die Station befindet. Der Radius ist dabei die ermittelte Distanz. Erst bei drei Stationen existiert nur noch ein Schnittpunkt [Röh08, S. 87 ff.].

Die Entfernung über ein Dämpfungsmodell zu bestimmen hat vor allem bei W-LAN, aber auch bei anderen Frequenzen entscheidende Nachteile. Mehrwegeausbreitung, Dämpfung durch Wände und andere Gegenstände verursachen einen Fehler beim Bestimmen der Distanz. Das Nutzen von anderen Techniken zur Ortung ist demnach dem RSSI Ansatz in Gebäuden vorzuziehen [Röh08, S. 88 f.].

2.1.3. AoA

Statt Entfernungen zu ermitteln, wird beim Angle of Arrival (AoA) Verfahren der Winkel zu einer Station gemessen. Dazu werden in der Regel an den festen Stationen mehrere Richtempfänger installiert. Bei zwei Stationen kann durch die Triangulation die Position des mobilen Gerätes bestimmt werden.

Das mobile Gerät befindet sich auf der Geraden, welche durch den Einfallswinkel beim Empfänger definiert ist. Durch zwei Geraden kann der eindeutige Schnittpunkt und somit die Position bestimmt werden. Dieses Verfahren benötigt einen höheren Hardwareaufwand als bei RSSI und ist demnach schlecht für den gewünschten Einsatz geeignet. Ein weiterer Nachteil ist, dass diese Methode nur funktioniert, wenn mobiles Gerät und Empfangsstation in direkter Sichtverbindung stehen. Ansonsten würde es zu Signalreflexionen kommen, welche den Einfallswinkel und somit die Ortungsgenauigkeit beeinflussen [Str+08, S. 44].

2.2. Fingerprintverfahren

Die Techniken um aus Messwerten direkt einen Standort abzuleiten, sind entweder sehr aufwendig mit Bezug auf die einzusetzende Hardware oder sind nicht genau. Ein anderes Verfahren zur Positionsermittlung sind so genannte Fingerprinttechniken. Dabei wird versucht jeden Ort durch ein charakteristisches Signalmuster wiederzuerkennen. Ein Fingerprintverfahren besteht aus zwei Phasen [Str+08, S. 45 f.]:

Offline-Phase Der Bereich, in dem geortet werden soll, wird zunächst in eindeutige Zellen aufgeteilt. Dabei spielt die Granularität der Zellen eine entscheidende Rolle bei der späteren Ortungsgenauigkeit. An diesen Stellen werden die charakteristischen Signalmuster aufgenommen und zusammen mit ihrer Position in einer Datenbank gespeichert [Str+08, S. 45].

Online-Phase In diesem Schritt soll die eigentliche Ortung durchgeführt werden. Dazu werden die gleichen charakteristischen Signale aufgenommen, wie innerhalb der Offline-Phase. Dieses Muster wird danach mit allen gespeicherten Fingerprints in der Datenbank verglichen [Str+08, S. 45].

Diese Möglichkeit der Ortung besitzt jedoch zwei grundlegende Nachteile. Zunächst müssen die charakteristischen Signale für jede einzelne Zelle aufgenommen werden. Je nach Granularität und Größe der Fläche stellt dies einen erheblichen Aufwand dar. Des Weiteren sind die aufgenommenen Signale von den Umgebungsbedingungen abhängig. Das bedeutet, dass eine Veränderung der Umgebung gleichermaßen zu anderen Signalen in dieser führt. Werden die Fingerprints nicht aktuell gehalten, verringert sich mit fortschreitendem Einsatz auch die Genauigkeit [Str+08, S. 45].

Die in den folgenden Abschnitten vorgestellten Ähnlichkeitsmaße dienen dazu, für einen Messwert den ähnlichsten aus der Fingerprintdatenbank zu suchen. Diese funktionieren als Vergleichsverfahren, um zwischen den Datenbankeinträgen eine Ähnlichkeitsreihenfolge im Bezug zu den Messdaten

zu erreichen. Ein Ähnlichkeitsmaß muss je nach Anwendungszweck ausgesucht werden. In dieser Arbeit werden drei verschiedene Maße auf ihre Tauglichkeit für den Einsatz untersucht.

2.2.1. Euklidische Distanz

Wird W-LAN für das Fingerprintverfahren genutzt, kann als charakteristisches Signal der RSSI genutzt werden. Dazu wird der RSSI für jeden vorhandenen Access-Point gespeichert. Daraus lässt sich ein Vektor ableiten, dessen Dimension die Anzahl der Access-Points ist [KK04].

Mittels euklidischer Distanz kann anschließend die Ähnlichkeit zwischen Messvektor und Fingerprint ermittelt werden. Dieses Distanzmaß ist wie folgt definiert:

$$d(\vec{x}, \vec{y}) = \|\vec{x} - \vec{y}\| = \sqrt{\sum_{i=1}^N (x_i - y_i)^2} \quad (2.2)$$

Als Beispiel soll ein Szenario dienen, bei dem drei Access-Points eingesetzt werden. Ein Fingerprint ist dann ein Punkt in einem dreidimensionalen Raum. Der Abstand zwischen Messvektor und Fingerprint ist die direkte Verbindung zwischen beiden Punkten. Werden mehr Access-Points eingesetzt, wird die Distanz in einem höher dimensionalen Vektorraum bestimmt [KK04].

Bei einer Lokalisierung kann anschließend die Distanz zu allen Fingerprints berechnet werden. Der Fingerprint mit der geringsten Distanz ist, nach diesem Maß, dem Messvektor am ähnlichsten. Der ähnlichste Fingerprint kann dann als Position angesehen werden, an dem die Messung stattfand. Eine Distanz von 0 bedeutet eine maximale Übereinstimmung. Dieses Maß wurde im Bereich der W-LAN Ortung eingesetzt. Es eignet sich gut, wenn die Messwerte in den einzelnen Dimensionen kaum zeitlichen Schwankungen unterliegt [GGH10].

2.2.2. Kosinus-Ähnlichkeit

Vergrößern sich mehrere Komponenten des Vektors proportional, steigt bei der euklidischen Distanz der Abstand zwischen den Vektoren an, obwohl die Vektoren zum Beispiel weiterhin in eine gleiche Richtung zeigen. Die Kosinus-Ähnlichkeit, welche beispielsweise im Gebiet des *Information Retrieval* genutzt wird, umgeht dieses Problem, indem es statt Abständen den Winkel zwischen zwei Vektoren verwendet [MRS08, S. 121 f.].

Der Kosinus zwischen zwei Vektoren kann durch die folgende Gleichung berechnet werden:

$$d(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\| \|\vec{y}\|} \quad (2.3)$$

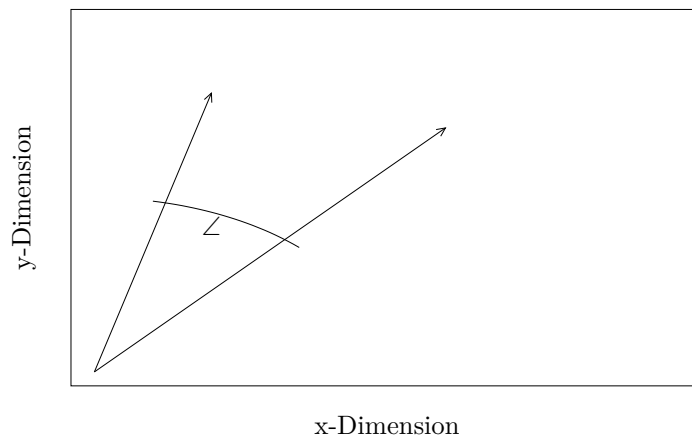


Abbildung 2.2.: Kosinus-Ähnlichkeit am Beispiel zweier Vektoren

Die Resultate dieser Operation liegen zwischen -1 und 1 . Ein Ergebnis von -1 bedeutet, dass beide Vektoren in entgegengesetzte und bei einem Ergebnis von 1 beide in die gleiche Richtung zeigen. Dieser Zusammenhang ist in Abbildung 2.2 dargestellt.

Dieses Maß, welches nicht von der Länge der Vektoren, sondern durch ihre Richtung beeinflusst wird, könnte beim Wiedererkennen von Magnetfeldern ebenso verwendet werden. Der Grundgedanke ist, dass verschiedene Objekte in einem Gebäude das natürliche Magnetfeld ablenken oder abschirmen. Wenn das Magnetfeld somit abgelenkt wird, passiert dies in eine charakteristische Richtung. In diesem Fall könnte der Vektorwinkel sich besser zum Vergleichen eignen als die euklidische Distanz.

2.2.3. Wahrscheinlichkeitsbasierter Vergleich

Die Sensoren, die die charakteristischen Signale aufnehmen, besitzen in der Regel einen Messfehler. Dieser führt zum Beispiel bei W-LAN dazu, dass die Signalstärke schwankt, obwohl der Messort nicht gewechselt wird. In der Regel passiert dies rundum den Erwartungswert, der „echten“ Signalstärke.

Dieser Messfehler verhält sich bei einigen Sensoren (siehe Abschnitt 3.2.6) wie eine Normalverteilung. Die wahrscheinlichkeitsbasierten Ähnlichkeitsmaße versuchen den Messfehler zu berücksichtigen und bei Sensoren, die einem Fehler unterliegen, ähnliche Werte als zusammengehörig zu erkennen.

Einer dieser, und in dieser Arbeit verwendete, Ansätze ist das gaußsche Ähnlichkeitsmaß, welches durch

$$d(\vec{x}, \vec{y}) = \prod_{i=1}^N \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x_i - y_i)^2}{2\sigma^2}\right) \quad (2.4)$$

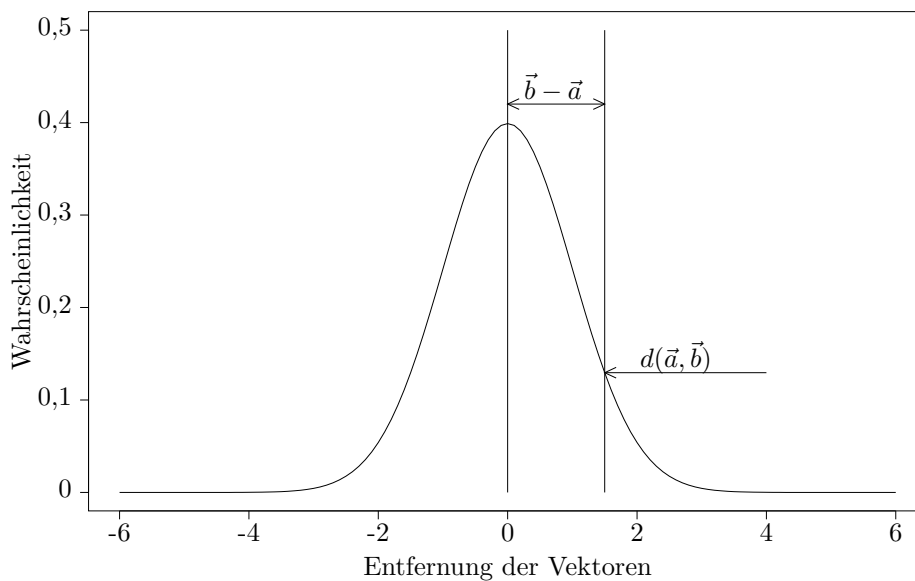


Abbildung 2.3.: Gaußsche Ähnlichkeit am Beispiel zweier eindimensionaler Vektoren bei $\sigma = 1$. Die Entfernung von \vec{a} und \vec{b} ist in diesem Fall 1,5.

definiert ist. Dabei sind \vec{x} und \vec{y} der gemessene Vektor und ein Fingerprint. N gibt die Anzahl der Dimensionen des Messwertes an. Die Standardabweichung σ gibt an, wie stark der Messwert streut. Dieser hängt von dem verwendeten Sensorsystem ab und muss empirisch bestimmt werden [Hon+, S. 3]. Dieses Maß ist in Abbildung 2.3 am Beispiel zweier Vektoren für eine Dimension dargestellt. Der Abstand der Vektoren definiert die daraus resultierende Ähnlichkeit.

Für jede Dimension des Messvektors wird bestimmt, wie ähnlich sich diese jeweils ist. Dazu wird die Differenz beider Werte genutzt. Diese wird verwendet um die Ähnlichkeit zueinander an der gaußschen Glocke zu bestimmen. Bei einer von Differenz 0 wird somit der Wahrscheinlichkeitswert am Maximum der Glockenfunktion genutzt. Größere Differenzen ergeben demnach eine geringere Ähnlichkeit. Beim Auswerten dieser Funktion ist der Fingerprint am ähnlichsten, der die größte Wahrscheinlichkeit besitzt.

Für andere Messfehlerverteilungen existieren ebenfalls andere Ähnlichkeitsmaße, welche in Honkavirta u. a. [Hon+] vorgestellt werden.

2.3. Technologien zur Ortung

Eine Standardlösung zur Positionierung außerhalb von Gebäuden ist das GPS. Dieses kann aber nicht, beziehungsweise nur schlecht, in Gebäuden eingesetzt werden, da Mauern und andere Hindernisse die Signale der Satelliten dämpfen. Mittels Standardempfängern in mobilen Geräten reicht die Signalstärke daher nicht aus, um Ortungen durchzuführen [ETZ05].

Die in Abschnitt 2.1 vorgestellten Verfahren können auf unterschiedliche Technologien angewandt werden. Dieser Abschnitt stellt den aktuellen Stand der Technik dieser Techniken dar.

2.3.1. W-LAN

Der Funkstandard IEEE-802.11, auch W-LAN genannt, wird immer mehr in Gebäuden, für den Zugang zu einem Netzwerk, eingesetzt. Dies spricht dafür W-LAN zu benutzen, da meist vorhandene Infrastruktur genutzt werden kann.

Um die in Abschnitt 2.1.2 vorgestellte Trilateration zu verwenden, muss aus dem RSSI eine Distanz vom mobilen Gerät zur Basisstation errechnet werden. Die Dämpfungsmodelle hängen von der verwendeten Ortungstechnologie und dem Einsatzort ab. Ein solches Modell für W-LAN wurde zuerst im RADAR Projekt vorgestellt [BP00].

Aufgrund von nicht vorhersehbaren Abschirmungen in Gebäuden ist es jedoch schwierig, ein Propagation Model aufzustellen. In Goldsmith [Gol05, S. 45] wird beispielsweise eines vorgestellt, welches Signalabschirmungen, die in Gebäuden auftreten, berücksichtigt. Es existiert bisher kein Modell, das alle Effekte, die ein Signal dämpfen, beachtet. Der Einsatz eines Propagation Models für die Lokalisierung mittels W-LAN führt demnach zu Ungenauigkeiten bei der Ortung [Au10, S. 14].

Der alternative Ansatz ist, ein Fingerprintverfahren zu benutzen. Wie in Abschnitt 2.2 beschrieben, kann durch Messen der Signalverhältnisse an einem Ort bei der Lokalisierung die Position besser geschätzt werden. Die Ortungsgenauigkeit hängt auch von der genutzten Zellengröße ab [KK04].

Die Signalverhältnisse können sich mit der Zeit an einem bestimmten Punkt durch Veränderungen in der Umgebung ändern. So beeinflussen neue Möbel oder andere elektrische Einflüsse, wie Bluetooth, welches im selben Frequenzband sendet, die Signalstärke. Die Folge ist, dass die Fingerprintkarte aktualisiert werden muss. In Widyawan [Wid10] und Chen u. a. [Che+05] wird versucht, die regelmäßige Aktualisierung der Karte zu vermeiden, indem diese automatisch erneuert wird.

Diese Aktualisierung basiert in Widyawan [Wid10] auf einer Pfadanalyse des mobilen Gerätes. Zunächst wird mittels Propagation Model eine Fingerprintkarte geschätzt. In der Online-Phase wird der Pfad des mobilen Gerätes verfolgt. Dabei entstehen durch die Messungenauigkeit mehrere mögliche Pfade welche anschließend auf ihre Wahrscheinlichkeit bewertet werden. Die während der Ortung aufgenommenen RSSI werden genutzt, um die Fingerprintkarte auf dem Pfad zu aktualisieren.

Zur Verfeinerung der geschätzten Positionen können weitere Sensorsysteme hinzugezogen werden [Che+05]. Dazu werden innerhalb eines Gebäudes zusätzliche RFID Referenztags platziert. Mittels Fingerprintverfahren und der weiteren Lokalisierungsquelle des RFID Systems, kann die Ortung verbessert und die Karte aktualisiert werden.

2.3.2. RFID

RFID, welches eigentlich zur Objektidentifikation genutzt wird, kann zur Ortung eingesetzt werden. Ein RFID System besteht in der Regel aus einem oder mehreren Readern und mehreren Tags. Reader sind, im einfachsten Fall, in der Lage die eindeutige Identifikationsnummer der Tags auszulesen. Die Lesereichweite variiert je nach System zwischen wenigen Zentimetern und mehreren Metern. Eine Ortung kann ähnlich wie bei W-LAN durchgeführt werden.

Ein anderer Ansatz wird in Bergemann und Sieck [BS11] vorgestellt. Dieser RFID basierter LANDMARC Algorithmus berechnet die Position eines Tags nicht nur über die Distanz zu den Readern, sondern verwendet noch mehrere Hilfstags. Diese werden zum Beispiel gitterartig an der Zimmerdecke befestigt. Der zu ortende Tag kann durch die Nähe zu den stationären Hilfstags genauer geortet werden. Der LANDMARC Ansatz bietet eine höhere Genauigkeit, jedoch besteht ein größerer Installationsaufwand [Ni+04].

Der Ansatz in Ozdenizci u. a. [Ozd+11] sieht vor, dass in einem Gebäude mehrere Tags platziert werden. Das mobile Gerät besitzt einen RFID Reader und kann an bestimmten Wegpunkten im Gebäude an die Tags gehalten werden. Die in diesem Szenario eingesetzte Technologie besitzt nur eine Reichweite von wenigen Zentimetern, deshalb kann beim Lesen eines Tags darauf geschlossen werden, dass der Reader sich unmittelbar in der Nähe befinden muss. Für diesen Moment kann eine sehr genaue Position ermittelt werden, jedoch ist es dabei nicht möglich, die Position zu verfolgen, wenn das mobile Gerät sich vom Tag entfernt.

2.3.3. Magnetfelder

Die Erde besitzt ein Magnetfeld, welches es ermöglicht über Kompass die Ausrichtung gen Norden zu bestimmen. Aktuelle mobile Geräte enthalten oft einen Magnetsensor, welcher zwei oder drei Achsen des Magnetfeldvektors bestimmen kann.

Blankenbach, Norrdine und Hellmers [BNH11] beschreibt ein Verfahren, bei dem künstliche Magnetfelder erzeugt werden. In dem vorgestellten System existieren mindestens drei Quellen, welche nacheinander an- beziehungsweise ausgeschaltet werden. Aus der Stärke und der Richtung des am mobilen Gerät empfangenen Magnetfeldes wird die Position des Gerätes berechnet. Ein weiteres Verfahren, welches künstliche niederfrequente Magnetfelder erzeugt, wird in Prigge [Pri04] beschrieben.

Bei der Selbstlokalisierung von Robotern spielen Kompass eine Rolle beim Bestimmen der Ausrichtung. In Gebäuden ist aber aufgrund von Stahlträgern und elektrischen Geräten das Erdmagnetfeld gestört. Diese Störung kann jedoch genutzt werden um eine Lokalisierung durchzuführen. Suksakulchai u. a. [Suk+00] beschreibt eine Ortung von einem Roboter auf einem Flur. Das veränderte Magnetfeld in dem Flur wird vor der Lokalisierung gemessen. Anschließend kann die charakteristische Ablenkung des Feldes wiedererkannt und daraus auf die Position im Flur geschlossen werden. Weitere Versuche existieren, bei denen eine Fingerprintkarte der Magnetfeldstörung aufgezeichnet

wurde. Diese Ansätze orten jedoch nur im eindimensionalen Fall oder auf einer kleinen Fläche mit wenigen Quadratmetern [ML98; Val+10; HK09; HK08].

Wird kein künstliches Feld erzeugt, sondern eine Datenbank über die Magnetfelder, bietet dies den Vorteil das im Grunde keine zusätzliche Infrastruktur installiert werden muss, da die zu messenden Signale bereits existieren.

2.3.4. Visuelle Marker

Schnelle Prozessoren und Kameras in mobilen Geräten ermöglichen die Ortung mittels visueller Marker. Dabei wird Bilderkennung genutzt, um markante Punkte in einer Umgebung zu erkennen. Ähnlich den Fingerprintverfahren bei W-LAN werden charakteristische Signale, in diesem Fall Bildsignale, aufgezeichnet. Bei der späteren Ortung werden kontinuierlich Bilder aufgenommen, daraus markante Punkte extrahiert und diese in einer Bilddatenbank gesucht [FG04].

Redžić u. a. [Red+09] beschreibt eine hybride Lokalisierung, bei dem mittels Funkortung eine grobe Position ermittelt wird. Anschließend wird über Bildauswertung die Positionsschätzung verbessert. Ein ähnliches Verfahren wird gleichfalls in Conaire u. a. [Con+08] beschrieben.

Der Einsatz von Bilderkennungsalgorithmen bietet den Nachteil, dass der Nutzer eine Kamera mitführen und auf die visuellen Marker richten muss. Ist diese Restriktion bei der Ortung von Robotern hinnehmbar, gestaltet sich dies bei menschlichen Nutzern schwierig.

2.3.5. Weitere Ansätze

Es existieren weitere Technologien zur Ortung. Einige davon verwenden nur andere Funkstandards, wie Bluetooth oder ZigBee, welche in Gebäuden zum Beispiel kaum Abschirmungseffekten unterliegen oder kostengünstiger sind als W-LAN [HNS03]. Bezogen auf das Argument, dass eine W-LAN Infrastruktur in Gebäuden oft vorhanden ist, stellen diese Varianten eher die technische Machbarkeit unter Beweis.

Ein aktuelles Mobiltelefon zum Beispiel enthält viel Sensorik, wie Beschleunigungssensor und Gyroskop. Damit ist es möglich Beschleunigungen zu messen und dadurch Schritte zu zählen. Über die Anzahl der Schritte und die Ausrichtung des Sensors kann anschließend der Verlauf verfolgt werden, was Pedestrian Dead Reckoning (PDR) genannt wird. Das Verfolgen der Bewegung liefert demnach keine eindeutige Position, sondern ist nur nützlich wenn dies mit anderen Ortungsergebnissen korreliert werden kann. Es wird folglich eine definierte Anfangsposition benötigt oder im Verlaufe der Ortung können andere Schätzergebnisse dadurch verfeinert werden. Ein Nachteil dieser zwei genannten Sensoren ist, dass sich kaum vermeidbare Messfehler mit der Zeit aufaddieren und somit sehr groß werden können. Es müsste infolge dessen regelmäßig die Berechnung der Änderung neu begonnen werden um weiterhin nutzbare Ergebnisse zu erzielen [ARL11; EM06; Wan+07b; Ang+10].

2.3.6. Zusammenfassung

Die Verwendung von W-LAN bietet den Vorteil, dass die benötigte Infrastruktur oft bereits vorhanden ist. Eine Ortung über den RSSI hat den entscheidenden Nachteil, dass bisher kein optimales Propagation Model gefunden wurde, welches alle Störfaktoren der Signalstärke berücksichtigt. Der Ansatz über das statistische Fingerprintverfahren erzielt gute Ergebnisse, wobei diese Karte regelmäßig aktualisiert werden muss, da sich Signalverhältnisse bei Umgebungsänderungen ändern. Ähnlich ist es bei aktiven RFID und Bluetooth Anwendungen, welche den gleichen Restriktionen wie W-LAN unterliegen.

Der LANDMARC Ansatz bietet eine gute Genauigkeit, ist aber unpraktikabel, da dafür viele Hilfstags installiert werden müssen.

Visuelle Marker für die Ortung zu verwenden lieferte gute Ergebnisse, was das Wiedererkennen von markanten Punkten anbelangt. Die benötigte Rechenzeit, der Aufwand zum Erstellen einer Bild-datenbank und die mangelnde Praktikabilität bei der Benutzung sprechen gegen die Verwendung auf einem mobilen Endgerät.

Ein künstliches Magnetfeld für die Ortung zu benutzen lieferte gute Ortungsergebnisse, scheitert jedoch an der Umsetzung. Beliebige Gebäude mit künstlichen Magnetfeldern auszustatten ist nicht praktikabel. Vielversprechender sind die Ansätze das bereits vorhandene Erdmagnetfeld zu nutzen. Diese Vorgehensweise benötigt keine zusätzliche Infrastruktur und kann in Gebäuden problemlos installiert werden.

Viele Forschungsprojekte beschäftigen sich aufgrund der Einschränkungen von einzelnen Technologien damit, verschiedene Ansätze miteinander zu kombinieren und somit die Nachteile gegeneinander auszugleichen.

2.4. Wahrscheinlichkeitsbasierte Positionsschätzung

Nahezu jedes System zur Lokalisierung besitzt unterschiedliche Probleme, welche meistens aus Messfehlern resultieren. Beispielsweise ist die Ortung mittels W-LAN und dem RSSI nicht exakt, da die gemessenen Signale durch Mehrwegeausbreitung oder Mauern gestört sind. Zur Verbesserung der Ortung werden somit Algorithmen zur Sensorfusion eingesetzt: diese kombinieren die ungenauen Positionen der einzelnen Systeme zu einem genaueren Standort.

Die genutzten Sensoren können in der Regel keine exakten Daten liefern, sondern enthalten Messfehler. Um die Position zu ermitteln werden wahrscheinlichkeitsbasierte Methoden genutzt, die die Fehler berücksichtigen und verkleinern.

Die nachfolgenden Verfahren basieren alle auf der Wahrscheinlichkeitstheorie, deshalb werden zunächst die Grundlagen dafür erläutert.

2.4.1. Wahrscheinlichkeitstheoretische Grundlagen

Grundgedanke der Positionsschätzung mit Hilfe der Wahrscheinlichkeitstheorie ist es, über ungenaue Messwerte einen Systemzustand zu schätzen. Wurde ein Zustand ermittelt und treffen neue Messwerte ein, kann auf Grund des letzten Zustands und den neuen Daten der neue Zustand besser geschätzt werden. Wahrscheinlichkeitsprozesse werden genutzt, um zum einen die fehlerhaften Messungen und zum anderen die daraus resultierenden ungenauen Positionen zu modellieren. Weiterhin ist es möglich, später folgende weitere ungenaue Messungen mit den vorherigen zu verbinden um dadurch zu einem besseren Ergebnis, das heißt eines mit einer geringeren Ungenauigkeit, zu gelangen. Um die später eingeführten Filter nachzuvollziehen, werden die notwendigen Begriffe und Notationen eingeführt [Obs09, S. 7].

Eine Zufallsvariable X kann bei einem Zufallsexperiment verschiedene Werte annehmen. Die Zuordnungsvorschrift $X(\omega)$ bildet das konkrete Versuchsergebnis auf eine reelle Zahl ab (siehe Formel 2.5). Die Ereignisse ω , aus der Menge Ω , hängen vom Zufall ab, somit ist ebenfalls die Funktion $X(\omega)$ vom Zufall abhängig. Zusätzlich wird noch zwischen kontinuierlichen und diskreten Zufallsvariablen unterschieden. Diskrete Variablen besitzen eine endliche Menge an Realisierungen (siehe [Bos11, S. 55 f.] und [Obs09, S. 7]).

$$X = \{x : x = X(\omega) \mid \omega \in \Omega, x \in \mathbb{R}\} \quad (2.5)$$

Die Wahrscheinlichkeit, das die Zufallsvariable X den Wert x annimmt, wird mit $P(X = x)$ bezeichnet [Bos11, S. 56].

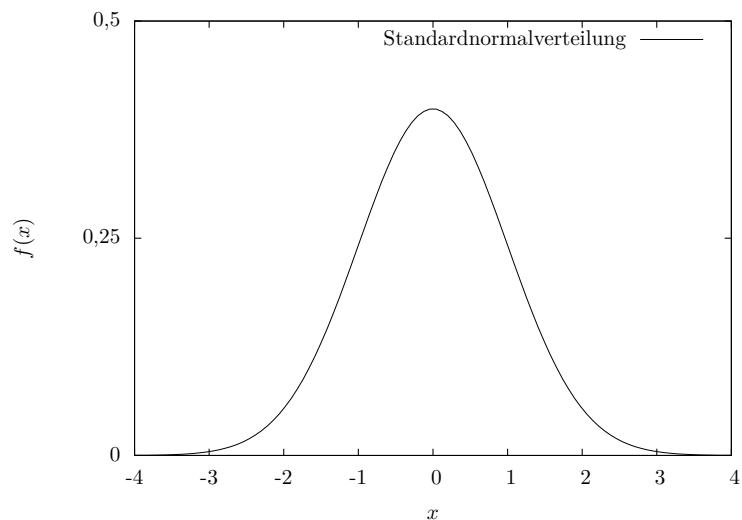
Wahrscheinlichkeitsdichte

Mit Hilfe der Verteilungsfunktion kann angegeben werden, mit welcher Wahrscheinlichkeit die Variable X nicht größer als das Ereignis x ist. Die Funktion ist definiert durch:

$$F(x) = P(X \leq x) \quad (2.6)$$

Die Wahrscheinlichkeitsdichtefunktion (WDF) ist die erste Ableitung der Verteilungsfunktion. Soll die Wahrscheinlichkeit errechnet werden, dass die Zufallsvariable in einem bestimmten Intervall liegt, wird die WDF über eben diesen integriert [Obs09, S. 7].

$$P(a \leq X \leq b) = \int_a^b f(x) dx \quad (2.7)$$

Abbildung 2.4.: Normalverteilung, $\mu = 0, \sigma^2 = 1$

Normalverteilung

Eine wichtige und in den Filtern oft genutzte Wahrscheinlichkeitsverteilung ist die Normalverteilung. Sie wird beispielsweise genutzt um das System- beziehungsweise Messrauschen zu modellieren, da ihre Verteilung dem jeweiligen Rauschen sehr nahe kommt [Obs09, S. 8].

Die Dichtefunktion der Normalverteilung ist durch

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right) \quad (2.8)$$

beschrieben. Wobei x der Zufallswert, σ^2 die Varianz und μ der Mittelwert ist [Fel68, S. 174]. In Abbildung 2.4 ist diese Funktion dargestellt. Eine Kurzschreibweise ist $\mathcal{N}(\mu, \sigma^2)$.

Bedingte Wahrscheinlichkeit

Die Wahrscheinlichkeit, dass die Zufallsvariable X den Wert x und Y den Wert y angenommen hat, wird Verbundwahrscheinlichkeit genannt und mit $P(X \cap Y)$ bezeichnet.

Soll ausgedrückt werden, dass $X = x$ unter der Voraussetzung das zuvor $Y = y$ gilt, wird die bedingte Wahrscheinlichkeit

$$P(X | Y) = P(X = x | Y = x) \quad (2.9)$$

$$= \frac{P(X \cap Y)}{P(Y)} \quad (2.10)$$

verwendet. Sind X und Y voneinander abhängig, so gilt der Satz von Bayes[Bos11, S. 29 f.]:

$$P(X | Y) = \frac{P(X \cap Y)}{P(Y)} = \frac{\frac{P(X \cap Y)}{P(X)} P(X)}{P(Y)} \quad (2.11)$$

$$= \frac{P(Y | X) P(X)}{P(Y)} \quad (2.12)$$

Diese Umformung der bedingten Wahrscheinlichkeit auf eine andere erfolgt, weil $P(Y | X)$ leichter zu berechnen ist, als $P(X | Y)$.

2.4.2. Markov Kette

Markov Ketten beschreiben ein Konzept, bei dem durch Kenntnis vorheriger Ereignisse auf zukünftige geschlossen werden kann. Der Zustand x_t kann durch die bedingte Wahrscheinlichkeit modelliert werden, wobei der Übergang von einem Zustand in einen anderen durch eine bestimmte Übergangswahrscheinlichkeit festgelegt ist. Bei einem PF wird angenommen, dass der zukünftige Zustand allein mit dem letzten Zustand vorhergesagt werden kann. Markov Ketten, bei denen nur der vorherige Zustand benötigt wird, heißen Markov Ketten erster Ordnung (siehe Formel 2.13) [Obs09, S. 12].

$$P(x_t | x_{t-1}, x_{t-2}, \dots, x_0) = P(x_t | x_{t-1}) \quad (2.13)$$

Eine Hidden Markov Kette beschreibt einen Prozess, bei dem der Zustand nicht direkt, sondern nur über Beobachtungen ermittelt werden kann. In Abbildung 2.5 ist ein solcher Prozess visualisiert, wobei z_t die Beobachtungen und x_t die Zustände sind [Obs09, S. 12 f.].

Bezogen auf die Positionsbestimmung bedeutet dies, dass die eigentliche Position nicht direkt bestimmt werden kann, sondern nur über Messungen zugänglich ist. Wird zur Positionsbestimmung beispielsweise GPS verwendet, ist die tatsächliche Position nicht zugänglich, sondern kann nur über die fehlerbehaftete GPS-Messung ermittelt werden. Am Beispiel von Fingerprintverfahren wird diese Unzugänglichkeit noch deutlicher. Werden RSSI Werte zur Ortung genutzt, existiert keine Position: erst per Lateration oder andere Verfahren kann auf eine Position geschlossen werden.

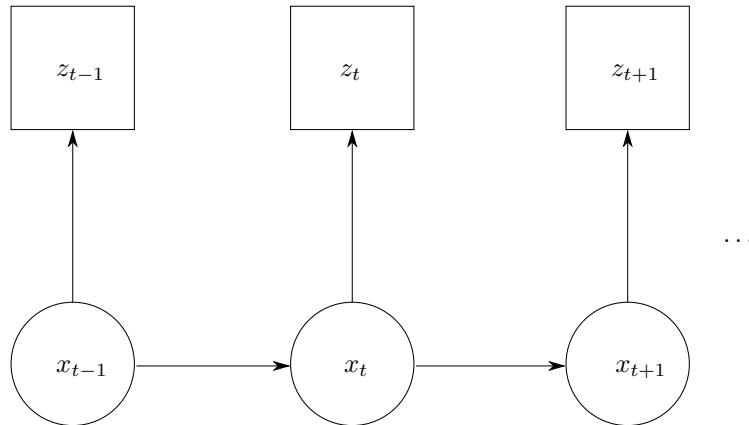


Abbildung 2.5.: Darstellung einer Hidden Markov Kette

2.4.3. Rekursive Bayes-Filter

Der Bayes-Filter wird zur Berechnung der Wahrscheinlichkeit $P(x_t | z_t)$, auch posteriori WDF genannt, genutzt. Die Erzeugung dieser Dichtefunktion erfolgt durch die Schritte *Vorhersage* und *Aktualisierung*. Die Vorhersage erfolgt durch die priori Wahrscheinlichkeit $P(x_t | z_{t-1})$: bei einer neuen Messung (Beobachtung) kann die Vorhersage dadurch aktualisiert werden [Wid10, S. 27].

Nach der Chapman-Kolmogorov-Gleichung kann die priori WDF durch

$$P(x_t | z_{t-1}) = \int p(x_t | x_{t-1})p(x_{t-1} | z_{t-1}) dx_{t-1} \quad (2.14)$$

dargestellt werden [Wid10, S. 27]. Bei Eintreffen einer neuen Messung z_t erfolgt die Aktualisierung durch:

$$P(x_t | z_t) = \frac{P(z_t | x_t)P(x_t | z_{t-1})}{\int P(z_t | x_t)P(x_t | z_{t-1}) dx_t} \quad (2.15)$$

Die Wahrscheinlichkeit $P(z_t | x_t)$ ist die Messwahrscheinlichkeit, $P(x_t | x_{t-1})$ die Übergangswahrscheinlichkeit von dem vorherigen Systemzustand zum aktuellen und $P(x_t | z_{t-1})$ ist die vorangegangene posteriori WDF [Wid10, S. 27].

Der rekursive Bayes-Filter nutzt zum Beispiel zur Berechnung der aktuellen Position, unter Berücksichtigung der neuen Messwerte, die zuletzt ermittelte Position.

2.4.4. Kalman-Filter

Der Kalman-Filter (KF) [Kál60] ist ein probabilistischer Filter, welcher auf der Theorie der Bayes-Filter aufbaut. Er ist eine Realisierung eines Hidden Markov Prozesses erster Ordnung und dient dazu, einen Systemzustand x , welcher nur über eine Messung z zugänglich ist, zu schätzen [Str+08, S. 89 f.].

Um dies zu erreichen, besitzt er ein Bewegungs- und ein Messmodell. Das Bewegungsmodell dient dazu, zukünftige Systemzustände zu schätzen. Das Messmodell bewertet diese Schätzung anhand von erhobenen Messdaten. Der KF nutzt ein lineares Modell und verwendet jeweils ein gaußverteiltes System- und Messrauschen [Str+08, S. 90].

Ein Nachteil des KF ist es, dass nur lineare Systemmodelle genutzt werden können. Darum wurden Erweiterungen geschaffen, welche es ermöglichen ebenso nicht lineare zu nutzen. Dazu gehören der Extended- und der Unscented KF [JU97].

Diese Filter werden in vielen Anwendungen, wie der Lokalisierung, eingesetzt. Die Gründe sind meist der Geschwindigkeitsvorteil gegenüber anderen Bayeschen Schätzern, da in einer Iteration nur eine Reihe von Matrizenoperationen durchgeführt werden müssen [FB11; Au10].

2.5. Partikel-Filter

Der PF, zuerst vorgestellt durch Gordon, Salmond und Smith [GSS93], versucht die Zustandsdichtefunktion des Systems durch eine endliche Menge an Partikeln N zu approximieren. Im Gegensatz zum KF muss die Änderung des Systems keinem linearen Modell unterliegen [Obs09, S. 26]. Weitere Namen für dieses Verfahren sind Bootstrap Filter, Condensation Algorithm, Interacting Particle Approximation und Survival of the Fittest [RAG04, S. 38 f.].

2.5.1. Grundidee

Die WDF (siehe Abbildung 2.6) stellt die Wahrscheinlichkeitsverteilung des Systemzustands dar. Ein Zustand kann beispielsweise die Position des mobilen Gerätes in einem Gebäude sein. Diese Dichtefunktion lässt es zu, dass die aktuelle Position uneindeutig ist. Die Abbildung 2.7 zeigt ein weiteres Beispiel der Dichtefunktion und wie diese durch die Partikel approximiert wird.

Jeder Partikel besitzt einen Zustand im Zustandsraum X und ein Gewicht. Diese werden in der Menge

$$\mathfrak{X} = \{\langle x_i, w_i \rangle \mid i = 1, \dots, N\} \quad (2.16)$$

zusammengefasst. Dabei ist x_i ein Zustand in X und w_i ein nicht negativer Wichtungsfaktor. Der Zustandsraum kann zum Beispiel den zu ortenden Bereich markieren, wobei ein konkreter Zustand

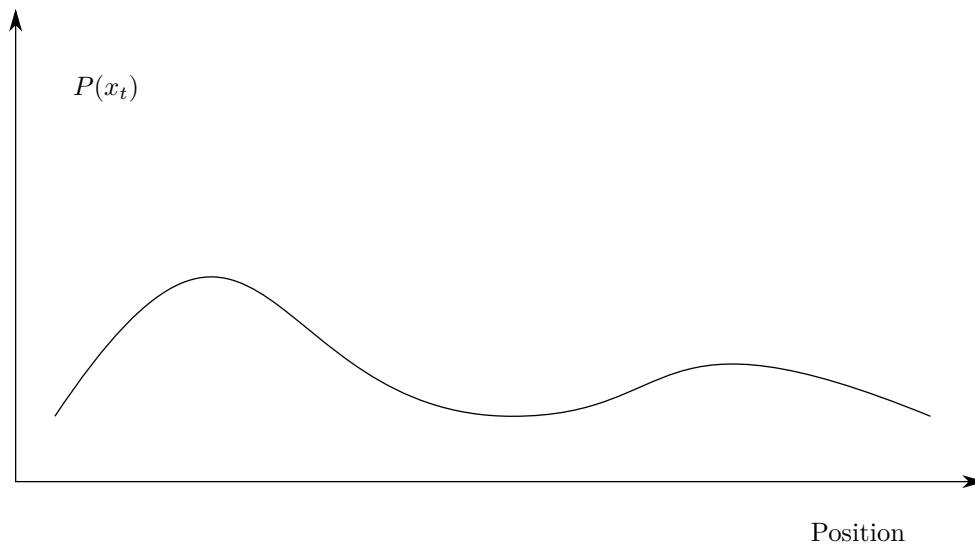


Abbildung 2.6.: Beispiel einer Dichtefunktion bei der Positionsermittlung

eine Position ist. Befinden sich mehrere Partikel in der Nähe zueinander, ist die Wahrscheinlichkeit, dass sich das System in dem jeweiligen Zustand befindet, groß [Obs09, S. 26].

Die Zustandsübergänge zum Zeitpunkt t lassen sich allgemein durch die nicht linearen Funktionen

$$x_t = f_{t-1}(x_{t-1}, n_{t-1}) \quad (2.17)$$

$$z_t = h_t(x_t, e_t) \quad (2.18)$$

beschreiben. Die Funktion 2.17 definiert den Zustandsübergang, basierend auf dem vorherigen Zustand und einem Systemrauschen. Formel 2.18 zeigt das Messmodell, wobei z_t der Messvektor und e_t das Messrauschen ist [Wid10, S. 25 ff.].

2.5.2. Darstellung der Dichtefunktion

Ziel des Partikel-Filters ist es, die normalerweise nicht integrierbare Dichtefunktion $p(x_t | z_t)$ zu approximieren (siehe Abschnitt 2.4.3). Diese lässt sich mit Hilfe der Partikel und ihrer Gewichte für einen bestimmten Zustand durch die Funktion

$$p(x_t | z_t) \approx \sum_{i=1}^N w_i \cdot \delta(x_t - x_{i,t}) \quad (2.19)$$

$$\sum_{i=1}^N w_i = 1 \quad (2.20)$$

darstellen [Wid10; Obs09]. Die Funktion $\delta(x)$ ist das Kronecker-Delta mit:

$$\delta(x) = \begin{cases} 1, & x = 0 \\ 0, & x \neq 0 \end{cases} \quad (2.21)$$

Die Dichtefunktion wird durch die Partikelverteilung bei kleinem N nur angenähert. Für $N \rightarrow \infty$ verschwindet der Approximationsfehler jedoch [Wen11, S. 182].

2.5.3. Allgemeiner Partikel-Filter

Die theoretischen Grundlagen für einen PF wurden in den vorherigen Abschnitten beschrieben. Diese können in den Algorithmus 2.1 überführt werden [Wid10, S. 29].

Algorithmus 2.1 Partikel-Filter mit Resampling-Schritt (\mathfrak{X}_{t-1}, z_t)

```

1:  $\bar{\mathfrak{X}}_t \leftarrow \emptyset$ 
2: for  $i = 1$  to  $N$  do
3:    $x_{i,t} \sim P(x_t | x_{i,t-1})$  ▷ Bewegungsmodell
4:    $w_{i,t} \leftarrow P(z_t | x_{i,t})$  ▷ Messmodell
5: end for
6:  $k \leftarrow \sum_{i=1}^N w_{i,t}$  ▷ Gesamtgewicht
7: for  $i = 1$  to  $N$  do
8:    $w_{i,t} \leftarrow k^{-1} w_{i,t}$  ▷ Normalisierung
9:    $\bar{\mathfrak{X}}_t \leftarrow \bar{\mathfrak{X}}_t + \langle x_{i,t}, w_{i,t} \rangle$ 
10: end for
11:  $\mathfrak{X}_t \leftarrow \text{Resample}(\bar{\mathfrak{X}}_t)$ 
12: return  $\mathfrak{X}_t$ 

```

Der Vorhersageschritt in Zeile 3 dient dazu, die Partikel im Zustandsraum anhand der Übergangswahrscheinlichkeit zu bewegen. Im Falle der Ortung werden die Partikel im Raum in ihrer Position verändert. Dieser Schritt wird in der Literatur *Motion Model*, nachfolgend nur noch Bewegungsmodell, genannt. Beispielsweise kann das Modell randomisiert sein, indem jeder Partikel in eine zufällige Richtung, unter Berücksichtigung der alten, bewegt wird [Wid10]. Andere Arbeiten nutzen einen Beschleunigungssensor um Schritte von Personen zu erkennen. Anhand von Entfernung und Winkel werden die Partikel entsprechend der erkannten Bewegung verschoben [Wan+07b; KW08].

Zeile 4 dient der Gewichtsanzpassung jedes Partikels. Dieser Schritt basiert auf dem *Measurement Model* beziehungsweise Messmodell; es bewertet, wie wahrscheinlich es ist, dass der jeweilige Partikel die aktuelle Messung erhalten hat. Wurde beispielsweise durch das Bewegungsmodell der Partikel in einen Zustand versetzt, bei dem die aktuelle Messung unwahrscheinlich ist, wird sein Gewicht reduziert. Dieser Schritt ist zentral, da durch ihn die WDF mit bestimmt wird [Wid10, S. 30].

Der Normalisierungsschritt dient dazu, das Gesamtgewicht aller Partikel auf 1 zu begrenzen; danach erfüllt die Partikelmenge die Bedingungen aus Formel 2.20.

Der letzte Schritt innerhalb der Iteration ist das *Resampling*, welches im Abschnitt 2.5.6 näher erläutert wird.

2.5.4. Bewegungsmodell

Ein Bewegungsmodell beschreibt den Übergang eines Zustands im System zu anderen. Genutzt wird es, um die Übergangswahrscheinlichkeit $P(x_t | x_{t-1})$ zu modellieren (siehe Algorithmus 2.1, Zeile 4). Dabei wird der Zustandsraum durch die Partikel erkundet. Das konkrete Modell hängt von dem Filterproblem ab. In diesem Abschnitt soll es für den Fall der Lokalisierung beschrieben werden [Wid10, S. 33].

In Widyawan [Wid10] wird ein Bewegungsmodell basierend auf zufälliger Richtungsänderung genutzt. Das bedeutet, dass jeder Partikel im Vorhersageschritt in eine zufällige Richtung bewegt wird.

Im dreidimensionalen Fall, besteht ein Systemzustand aus der Position im Raum, modelliert durch die x- und y-Koordinaten und der Etage z. Während der Vorhersage werden zwei Bewegungsparameter bestimmt. Zum einen die neue Geschwindigkeit, zum anderen die Richtung. Das Bewegungsmodell kann durch

$$x_{i,t} = \begin{pmatrix} p_{i,t}^x \\ p_{i,t}^y \\ p_{i,t}^z \end{pmatrix} = \begin{pmatrix} p_{i-1,t}^x + v_{i,t} \cos(\alpha_{i,t}) \Delta t + \eta_t \\ p_{i-1,t}^y + v_{i,t} \sin(\alpha_{i,t}) \Delta t + \eta_t \\ p_{i-1,t}^z + \Psi(a_t, a_{t-1}) \end{pmatrix} \quad (2.22)$$

beschrieben werden. Wobei v_t die Geschwindigkeit, α_t die Bewegungsrichtung, Δt die Zeit seit des letzten Vorhersageschritts und η_t das Systemrauschen ist. Die Bewegung unter den Etagen benötigt kein Messrauschen. Diese Dimension wird in Gebäuden genutzt um die Etage zu repräsentieren, es handelt sich somit um diskrete Werte [Wid10, S. 36]. Die Ermittlung der einzelnen Parameter und die Einbeziehung der dritten Dimension wird in Abschnitt 3.2.5 näher beschrieben.

Dieses Bewegungsmodell lässt es zu, die Bewegung eines Partikels über die natürlichen Parameter der Strecke und Richtung zu modellieren. Die einzelnen Parameter können, wie später erläutert, durch stochastische Prozesse gebildet werden.

In Klingbeil und Wark [KW08] werden die einzelnen Parameter nicht durch Zufallsverteilungen, sondern mittels Messungen bestimmt. Dazu wurden die Schritte und Schrittweite gezählt. Dieses Verfahren hat den Vorteil das die Partikel sich in die tatsächliche Richtung des Systemzustands bewegen, wobei es weniger falsch lokalisierte Partikel geben sollte. Ein Nachteil ist hingegen der erhöhte Hardwareaufwand.

2.5.5. Messmodell

Innerhalb des Partikel-Filters wird das Messmodell genutzt um die Wahrscheinlichkeit $P(z_t | x_t)$, im Algorithmus 2.1, Zeile 4, zu beschreiben. Es gibt an, wie wahrscheinlich es ist, dass ein bestimmter Partikel die aktuelle Messung erhalten hat. Mit der resultierenden Wahrscheinlichkeit wird der Wichtungsfaktor jedes Partikels bestimmt [Wid10, S. 37 f.]. Ähnlich dem Bewegungsmodell ist dieses vom Anwendungsfall abhängig.

Das Messmodell ist, bei Verwendung von mehreren Positionierungssystemen, gleichfalls die Komponente um die verschiedenen Positionen zusammen zu führen.

In Widyawan [Wid10] werden zur Lokalisierung zum einen W-LAN Fingerprints, aber ebenso Positionen über Ultraschall und RSSI basierte Distanzmessungen genutzt. Auch [KW08] nutzt die Positionierung über die Nähe zu einem Funkgerät. Fingerprinttechniken werden weiterhin in [KK04] eingesetzt.

Neben den Informationen des Sensors kann ebenfalls Wissen aus der Umgebungskarte genutzt werden, um Partikelbewegungen zu bewerten. Bewegt sich ein Partikel aufgrund des Bewegungsmodells durch eine Wand, kann dies keine gültige Verschiebung im Zustandsraum darstellen. Dies kann durch folgende Gleichung abgebildet werden [Wid10; KW08]:

$$w_{i,t} = \begin{cases} 0, & \text{wenn Wand durchquert} \\ P(z_t | x_{i,t}), & \text{sonst} \end{cases} \quad (2.23)$$

2.5.6. Verhindern von Partikeldegeneration durch Resampling

Da die Dichte $P(x_t | z_t)$ normalerweise nicht direkt ermittelt werden kann, wird stattdessen eine ähnliche Dichtefunktion genutzt. Dies wird in der Literatur *Importance Sampling* genannt. Bei diesem Verfahren kommt es jedoch zur Degeneration der Partikelmenge [DGA99, S. 198 f.]. Das bedeutet, dass nach einer geringen Anzahl an Iterationsschritten entweder die Partikel ein Gewicht nahe 0 oder nahe 1 haben. Die meisten Partikel tragen deshalb kaum zur Dichtefunktion bei, wohingegen wenige durch ihr hohes Gewicht diese dominieren. Weiterhin ist ersichtlich, dass dieser Effekt beim Importance Sampling nicht vermeidbar ist [DGA99, S. 198 f.].

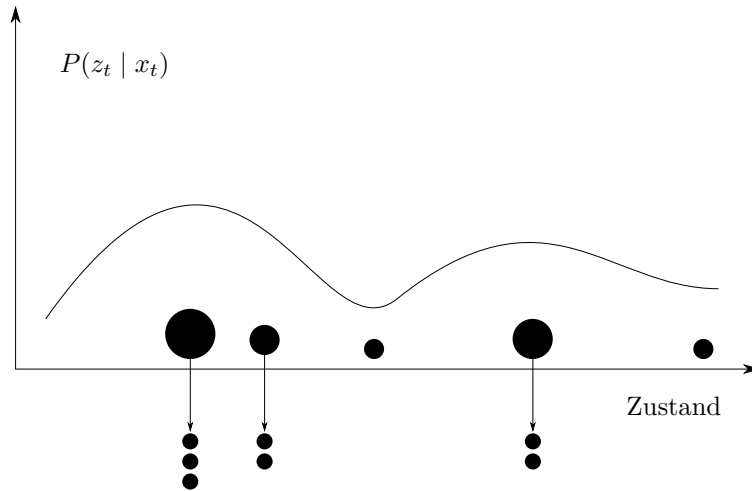


Abbildung 2.7.: Resampling Vorgang zur Vermeidung der Partikeldegeneration [nach RAG04, S. 44]

Um diese Degeneration aufzulösen, wurde der so genannte Resampling-Schritt eingefügt. Dabei werden Partikel mit geringem Gewicht aus der Verteilung entfernt und Partikel mit hohem Gewicht dupliziert. Anschließend werden die Gewichte normalisiert, sodass gilt:

$$\forall w_{i,t} = \frac{1}{N} \quad (2.24)$$

Die WDF wird durch diesen Schritt kaum verändert, da die entfernten Partikel kaum zur Dichte beigetragen haben. Die Abbildung 2.7 stellt den Resampling-Schritt grafisch dar [Obs09, S. 37 f.].

Ein möglicher Resampling Algorithmus wurde in [Wid10, S. 31] vorgestellt und ist in Algorithmus 2.2 erläutert.

Der Algorithmus ermittelt aus der nach Gewicht aufsteigend sortierten Liste der Partikel für jeden, ob dieser in der nächsten Iteration weiterhin genutzt werden soll, oder ob er durch einen geeigneteren ersetzt wird. In Zeile 3 wird eine gleichverteilte Zufallszahl ermittelt. In der inneren Schleife werden die Partikelgewichte aufaddiert. Übersteigt s die zuvor ermittelte Zufallszahl, wird der gerade ausgewählte Partikel in die Folgeliste aufgenommen.

Der Grundgedanke ist, dass Partikel mit geringem Gewicht kaum zur Inkrementierung von s beitragen, und folglich ihre Wahrscheinlichkeit in die neue Liste aufgenommen zu werden gering ist. Hingegen sorgen Partikel mit hohem Gewicht für ein Überschreiten von s ; diese werden anschließend in die Liste aufgenommen [Wid10, S. 30 ff.].

Weitere Algorithmen und ihre Bewertung werden in [RAG04, S. 42] und [Obs09, S. 40 ff.] vorgestellt. Die verschiedenen Varianten unterscheiden sich in ihrer Komplexität und der Varianz der neuen Partikelmenge.

Algorithmus 2.2 Resampling Verfahren ($\bar{\mathfrak{X}}_t$)

Require: $\bar{\mathfrak{X}}_t$ aufsteigend sortiert

```
1:  $\mathfrak{X}_t^* \leftarrow \emptyset$ 
2: for  $i = 1$  to  $N$  do
3:    $r \leftarrow$  Zufallszahl aus  $[0, 1]$ 
4:    $s \leftarrow 0$ 
5:   for  $j = 1$  to  $N$  do
6:      $s \leftarrow s + w_{i,t}$ 
7:     if  $s \geq r$  then
8:        $x_t^{*j} \leftarrow x_{i,t}$ 
9:        $w_t^{*j} \leftarrow \frac{1}{N}$ 
10:       $\mathfrak{X}_t^* \leftarrow \mathfrak{X}_t^* + \langle x_t^{*j}, w_t^{*j} \rangle$  ▷ Kopieren des ausgewählten Partikels
11:      break
12:     end if
13:   end for
14: end for
15: return  $\mathfrak{X}_t^*$ 
```

2.5.7. Ermitteln des Systemzustands

Alle Partikel approximieren die WDF, jedoch soll in den meisten Fällen ein konkreter Systemzustand ermittelt werden. Hierfür dient die Formel 2.19, welche die Wahrscheinlichkeit angibt, dass sich das System in einem bestimmten Zustand befindet. Der vom System geschätzte Zustand wird bestimmt, indem die Wahrscheinlichkeit für jeden einzelnen Zustand errechnet und anschließend der wahrscheinlichste ausgewählt wird.

Eine andere Möglichkeit ist in Obst [Obs09, S. 44] vorgestellt. Diese sieht vor nur den Partikel mit dem höchsten Gewicht zu wählen. Der Systemzustand entspricht dann dem Zustand des Partikels, wobei gilt:

$$x_t^{\text{System}} = x_{i_{\max},t} \quad i_{\max} = \max(w_{i,t} \mid i = 1, \dots, N) \quad (2.25)$$

2.6. Navigation in Gebäuden

Aufbauend auf der Position in einem Gebäude können Dienste wie Navigation in diesem angeboten werden. Im Folgenden Abschnitt wird der aktuelle Stand der Technik zur Navigation in Gebäuden aufgeführt und zugrundeliegende Algorithmen beschrieben.

2.6.1. Stand der Technik

Mit Navigationsgeräten in Fahrzeugen sind heutzutage die meisten Fahrer vertraut. Es gibt zahlreiche Geräte mit umfangreichem Kartenmaterial und Zusatzfunktionen. Die Ortung erfolgt in

der Regel über GPS. Die Algorithmen zum Finden des kürzesten Weges zwischen zwei Punkten sind oft Erweiterungen des bzw. direkt der Algorithmus von Dijkstra[Dij59]. Eine Variante ist beispielsweise der A^* -Algorithmus.

Die Algorithmen zum Finden des kürzesten Weges sind innerhalb von Gebäuden nicht anders. Besondere Anforderungen werden hingegen an die Visualisierung des Weges und an die Technik zur Ortung gestellt.

Gilliéron und Merminod [GM03] beschreibt die Implementierung eines Navigationssystems für Gebäude. Zur Ortung werden GPS und PDR genutzt. Beim Speichern der topologischen Innenkarte wird auf die Unterscheidung von verschiedenen Pfaden, wie Treppengänge, geachtet. Zur Verbesserung der Ortungsgenauigkeit können Karteninformationen genutzt werden um dadurch unwahrscheinliche Positionierungsergebnisse zu korrigieren.

Wang u. a. [Wan+07a] beschreibt eine hierarchische Gebäudestruktur. In der untersten Stufe werden Wände, Punkte und die Verbindungen unter diesen modelliert, die nächste Stufe beschreibt die Zusammensetzung von Räumen, Fluren und zum Beispiel Treppenaufgängen. Die semantische Ebene fügt diesen Informationen Bedeutung für den Endnutzer hinzu, wie die Namen der Personen in Räumen beziehungsweise ihre Funktionen. Diese Karteninformationen werden auf allen Ebenen der Navigation und Ortung genutzt; beispielsweise zur Verbesserung der Ortungsgenauigkeit, dem eigentlichen Berechnen der kürzesten Route und der Ausgabe von Navigationsanweisungen für den Nutzer.

In Butz u. a. [But+01] wurden verschiedene Darstellungsformen der Anweisungen für den Nutzer, basierend auf den technischen Ausstattungen der Endgeräte, diskutiert. Abhängig von der Ortungsgenauigkeit können unterschiedliche Kartenvisualisierungen angeboten werden. Als Beispiel wird angeführt, dass bei ungenauer Ortung eine große Übersichtskarte angezeigt bei genauerer Ortung hingegen eine Detailkarte visualisiert wird. Falls neben der Position auch die Ausrichtung des Gerätes bekannt ist, kann die Visualisierung um dreidimensionale Elemente erweitert werden um zum Beispiel Treppenaufgänge gesondert zu kennzeichnen.

Kontextbasierte Navigation und die Möglichkeit des Hinzufügens von Zusatzinformationen für die Navigation werden in Huang u. a. [Hua+09] diskutiert. In dieser Arbeit wird erläutert wie durch Nutzerfeedback die Navigationsanweisungen verbessert werden können. Hat der Benutzer beispielsweise Schwierigkeiten eine Anweisung zu verstehen, kann er dem System Feedback dazu geben. Dadurch können die Karteninformationen angepasst werden. Weiterhin ist es dem Nutzer in dem Szenario möglich besondere Plätze, zum Beispiel gute Einkaufsmöglichkeiten, zu markieren, sodass spätere Nutzer von diesen Informationen profitieren können.

Die zuvor genannten Navigationslösungen beschäftigen sich mit Ansätzen, die in der Regel nicht für sehbehinderte Menschen geeignet sind. Im Gegensatz dazu, wird in Swobodzinski und Raubal [SR08] ein Navigationssystem für Menschen mit Sehbehinderung entwickelt. In der Karte und den verwendeten Algorithmen werden Hindernisse berücksichtigt beziehungsweise Pfade erstellt, die für Blinde Orientierungspunkte bieten.

2.6.2. Umsetzung der Navigation

Um innerhalb eines Gebäudes navigieren zu können, muss die Struktur dessen für einen Algorithmus zugänglich gemacht werden. Um von einem Ort zu einem beliebigen anderen zu navigieren, wird ein Algorithmus zum Finden des kürzesten Pfades zwischen einem Startknoten und einem Endknoten benötigt. Zunächst wird kurz erläutert, was ein Graph ist und anschließend wird ein Algorithmus zur Berechnung des kürzesten Pfades beschrieben.

Definition eines Graphen

Ein Graph $G = (V, E)$ ist zunächst definiert durch ein Tupel von Knoten V und Kanten E . Die Darstellung von Knoten erfolgt in der Literatur oft über Kreise mit einer Beschreibung für diesen. Die Menge der Kanten beschreibt die Verbindung zwischen zwei Knoten. Eine Kante wird durch ein Tupel (u, v) dargestellt, wobei $u, v \in V$ [Cor+04, S. 1082].

Es existieren zwei grundlegende Arten von Graphen, zum einen gerichtete und zum anderen ungerichtete Graphen. In gerichteten Graphen ist eine Kante eindeutig in ihrer Richtung bestimmt. Die Kante (u, v) geht dementsprechend vom Knoten u zu v . In ungerichteten Graphen existiert diese Ordnung nicht, denn beide Richtungen sind möglich [Cor+04, S. 1082 f.].

Eine weitere Unterscheidung gibt es bei Graphen mit gewichteten beziehungsweise ungewichteten Graphen. In dieser Arbeit werden nur gewichtete genutzt. Demnach besitzt die Kante zwischen zwei Knoten ein Gewicht.

Algorithmen

Der Algorithmus von Dijkstra berechnet ausgehend von einem Startknoten die kürzeste Verbindung zu allen anderen Knoten im Graphen. Zunächst müssen einige formale Grundlagen für diesen Algorithmus festgelegt werden.

Ein Graph dient zur Repräsentation des Gebäudes und dessen Wege. Für den Dijkstra-Algorithmus wird ein gerichteter Graph mit nicht negativen Kantengewichten benötigt. Die Funktion $w(u, v)$ gibt das Gewicht zwischen den Knoten u und v zurück [Cor+04, S. 598].

Der Algorithmus 2.3 zeigt das Verfahren zur Bestimmung des kürzesten Weges zu allen Knoten im Graphen.

Die Zeilen 1-5 initialisieren den Algorithmus. Die Liste $d[v]$ speichert die kürzeste Distanz vom Startknoten s bis zum Knoten v . Anfänglich sind alle Distanzen auf unendlich gesetzt. $\pi[v]$ beschreibt den Vorgänger von v . Dies ist der Knoten über den man auf kürzestem Pfad v erreicht. Die Menge S enthält alle Knoten, für die bisher die kürzeste Strecke von s aus gefunden wurde. Q enthält alle noch zu besuchenden Knoten [Cor+04, S. 598 f.].

Zur Speicherung der offenen Knoten in Q kann für ein besseres Laufzeitverhalten eine *Priority-Queue* genutzt werden, wobei der Sortierschlüssel ein Eintrag aus der Liste d ist.

Algorithmus 2.3 Dijkstra-Algorithmus (G, s)

```
1: for all  $v \in V$  do ▷ Initialisierung
2:    $d[v] \leftarrow \infty$ 
3:    $\pi[v] \leftarrow \text{NULL}$ 
4: end for
5:  $d[s] \leftarrow 0$ 
6:  $S \leftarrow \emptyset$ 
7:  $Q \leftarrow v \in V$ 
8: while  $Q \neq \emptyset$  do
9:    $u \leftarrow \text{Entnimm-Min}(Q)$ 
10:   $S \leftarrow S \cup \{u\}$ 
11:   $Q \leftarrow Q \cap \{u\}$ 
12:  for all  $v \in \text{Adj}(u)$  do
13:    if  $d[v] > d[u] + w(u, v)$  then ▷ Relax Funktion
14:       $d[v] \leftarrow d[u] + w(u, v)$ 
15:       $\pi[v] \leftarrow u$ 
16:    end if
17:  end for
18: end while
```

Zeile 9 nimmt den Knoten aus der Prioritätsliste mit der aktuell kürzesten Distanz vom Start zum jeweiligen Knoten. Für diesen Knoten werden anschließend alle benachbarten Knoten besucht und entschieden, ob die Distanzliste angepasst werden muss [Cor+04, S. 598 f.].

Die Relaxionsfunktion in Zeile 13-16 prüft für jeden Nachbarn, ob über die Verbindungskante zwischen u und v eine kürzere Verbindung erreicht werden kann, um danach die Distanz in $d[v]$ anzupassen. Falls diese Distanz kürzer ist, wird sowohl die Distanz angepasst als auch ein neuer Vorgänger in $\pi[v]$ gespeichert. Am Ende des Algorithmus sind in d die kürzesten Wegstrecken und in π die passenden Vorgänger zum jeweiligen Knoten hinterlegt [Cor+04, S. 598 f.].

Um den kürzesten Pfad zu rekonstruieren, muss ausgehend vom Zielknoten die π Liste rückwärts zum Startknoten durchgegangen werden.

Eine Erweiterung des Dijkstra-Algorithmus ist der A^* -Algorithmus. Statt beim Durchsuchen des Graphen nur mit Hilfe der Kostenfunktion zum nächsten Knoten die Suche fortzusetzen, verwendet der A^* eine Heuristik. Dadurch ist es möglich die Suche zielgerichteter durchzuführen und nur die Knoten zu betrachten, die am wahrscheinlichsten an das Ziel führen [HNR68]. Beide genannten Algorithmen sind optimal, das heißt sie liefern den kürzesten Pfad im Graph. Der Dijkstra-Algorithmus ist ein oft genutzter Algorithmus für dieses Problem. Aufgrund der geringen Größe des Graphen im Gebäude, kann dieser genutzt werden, obwohl der A^* -Algorithmus bei größeren Graphen effizienter wäre.

3. Entwicklung einer Ortungsstrategie

Die Bestimmung des Ortes eines mobilen Gerätes ist zwingende Voraussetzung für eine Navigation. In diesem Kapitel wird beschrieben, mit welchen Verfahren zum einen die Ortung stattfinden kann und zum anderen wie nacheinander erfolgte Positionsmessungen genutzt werden können um die Standortinformation zu verbessern.

Das Kapitel über die Ortungsgrundlagen hat eine Übersicht über existierende Ortungstechnologien gegeben. In diesem Kapitel wird entsprechende Sensorik für diese Aufgabe ausgewählt und bewertet. Zum Zusammenführen von Ortungsinformationen wird weiterhin ein Filter ausgewählt und auf den vorliegenden Anwendungsfall adaptiert.

Ein mobiles Gerät hat in der Regel begrenzte Hardwareressourcen zur Verfügung, deshalb ist es notwendig die Wahl des Filter-Algorithmus darauf abzustimmen. Die ausgewählten Sensoren werden dahingehend überprüft ob sie sich eignen in einem Gebäude eingesetzt werden zu können.

3.1. Auswahl der verwendeten Sensorik

Im Abschnitt zum Stand der Technik, wurde gezeigt, dass eine Ortung über Magnetfelder realisiert werden kann (siehe Abschnitt 2.3.6). Das in Gebäuden gestörte Erdmagnetfeld wird dabei genutzt, um Referenzpunkte wiederzuerkennen. Die bisherigen Versuche anderer, bezogen sich entweder auf die Ortung im eindimensionalen Fall, wie einem Flur, oder nur auf eine sehr kleinen Fläche. Alle bisher vorgestellten Technologien, bis auf Magnetfelder, benötigen zum Funktionieren eine mehr oder weniger aufwändige Infrastruktur. Um die Ortung weiter zu verbessern, soll weitere Sensorik eingesetzt werden.

Die in diesem Abschnitt vorgestellten Methoden kommen ohne zusätzliche Installationen im Gebäude aus. Dazu gehören zum einen Lichtfingerprints zum präzisieren der Ortung im zweidimensionalen Raum und die Auswertung des Luftdruckes um eine Bestimmung der Etage zu gewährleisten.

Weiterhin sind die experimentellen Ergebnisse für eine Magnetfeldortung aufgeführt. Die Eignung von Lichtquellen zur Verbesserung der zweidimensionalen Ortung, sowie des Barometers zur Erkennung der Etage wird zunächst theoretisch erörtert.

Raum	Größe (in m ²)	Referenzpunkte
2.2.01	57,70	12
2.2.18	49,10	8
2.3.01	28,89	5
2.2.02	28,89	5

Tabelle 3.1.: Maße der Versuchsräume

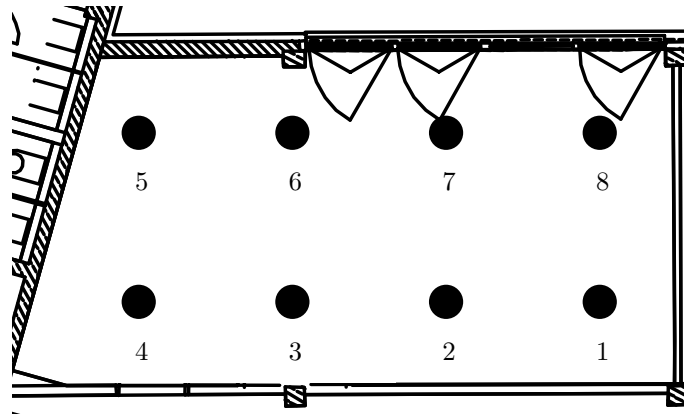


Abbildung 3.1.: Verteilung der Mess- und Referenzpunkte im Raum 2.2.18

3.1.1. Versuchsaufbau

Die Messungen fanden in den Büroräumen des Technologie- und Gründerzentrums Spreekie in Berlin statt. Insgesamt wurden vier Räume in der zweiten und dritten Etage untersucht. In Tabelle 3.1 sind die Maße sowie die Anzahl der erfassten Datenpunkte gelistet. Vor allem die aufgenommenen Magnetfelder hängen von den im Raum befindlichen elektronischen Geräten und ihrer Größe ab, dazu wurden Räume ausgesucht die eine unterschiedliche Aufteilung mit Möbeln und elektrischen Geräten bieten. Die Messungen wurden jeweils bei an- bzw. ausgeschalteten Geräten durchgeführt.

Die Aufzeichnung der Referenzpunkte und der Messwerte zum Lokalisieren fand am 8. November 2011 statt. Weiterhin galt es zu ermitteln, ob die vorhandenen Magnetfelder über längere Zeit stabil sind, folglich wurden Messwerte für diese Überprüfung erneut am 16. November aufgenommen.

Die Abbildung 3.1 zeigt die Messpunkteverteilung in einem der Büroräume.

Sowohl die Magnetfelder als auch die Lichtintensitäten wurden mittels Motorola Xoom² mit der Android Version 3.2 aufgenommen.

²vgl. <http://www.motorola.com/Consumers/DE-DE/Consumer-Products-and-Services/Android-Tablets/ci.MOTOROLA-XOOM-DE-DE.alt>, Abruf 16. Februar 2012

3.1.2. Ortung durch Wiedererkennen von Magnetfeldern

Messvektoren im Weltkoordinatensystem

Der über das Android Betriebssystem gemessene Magnetfeldvektor beschreibt das Feld relativ zum Koordinatensystem des Gerätes. Während der Lokalisierung ist jedoch nicht gewährleistet, dass der Sensor immer die gleiche Lage hat. Das Android Betriebssystem kann aus Beschleunigungs- und Magnetfeldsensoren die Orientierung des Gerätes im Raum bestimmen. Mittels der daraus erstellten Drehmatrix wird der gemessene Magnetfeldvektor in das Weltkoordinatensystem überführt. Innerhalb des Weltkoordinatensystems zeigt der positive Teil der x-Achse nach Osten, der y-Teil Richtung Nordpol und die z-Achse in den Himmel.³ Die in das Weltkoordinatensystem gedrehten Messvektoren werden fortan als normierte Vektoren bezeichnet.

Die Drehmatrix enthält jedoch aufgrund der Magnetfeldstörung selbst einen Fehler; vor allem die x- und y-Komponenten des Magnetfeldvektors können nicht korrekt in das Weltkoordinatensystem gedreht werden. Für jeden Referenzpunkt wurden vier Magnetfeldvektoren aufgenommen [Au10, S. 30]. Der Sensor wurde dabei jeweils in alle vier Himmelsrichtungen gedreht. Der in der Datenbank erfasste Referenzpunkt besteht aus dem Durchschnitt der zuvor gemessenen Vektoren.

Ergebnisse der Lokalisierung

Um von einem Messwert auf die Position schließen zu können, wurde für jeden Messwert die ähnlichsten Referenzpunkte ermittelt. Um die Ähnlichkeitsreihenfolge herzustellen, wurden die drei Ähnlichkeitsmaße aus Abschnitt 2.2 genutzt. Diese waren die euklidische Distanz, die Kosinus-Ähnlichkeit und das gaußsche Ähnlichkeitsmaß. Bei der Auswertung wurde es als Treffer gewertet, wenn der passende Referenzpunkt unter den zwei ähnlichsten Ergebnissen war.

In Tabelle 3.3 ist ein Auszug der Messwerte für den Büroraum 2.2.18 bei eingeschalteten Geräten dargestellt. Die Mess- und Referenzpunkte wurden nach dem Standort im Raum durchnummeriert. Der Punkt M1 ist demnach der Messwert welcher an der Stelle 1 aufgenommen wurde und der Punkt P1 entsprechend der Referenzvektor von der selben Position. Bei der Untersuchung wurden ebenfalls Messwerte zwischen den eigentlichen Referenzpunkten aufgenommen. Diese sind zum Beispiel mit M2-7 gekennzeichnet. Dies ist ein Messvektor, welcher zwischen den Stellen 2 und 7 aufgenommen wurde. Um die erzielten Ähnlichkeitswerte zu verifizieren, wurde in Tabelle 3.2 die Feldstärken an den jeweiligen Punkten aufgeführt. Die gaußsche Ähnlichkeit nutzt bei der Berücksichtigung von Messfehlern eine Standardabweichung welche für das beschriebene Szenario experimentell bestimmt wurde. Die Abweichung lag bei zirka $0,53 \mu\text{T}$.

An dieser Stelle sei auf den Messvektor M2-7 verwiesen, bei dem die ähnlichsten Referenzpunkte P2 und P7 waren. Zur Veranschaulichung dieses Messpunktes zwischen zwei Referenzvektoren, sei nochmals auf Abbildung 3.1 verwiesen.

³vgl. <http://developer.android.com/reference/android/hardware/Sensor.html>, Abruf 16. Februar 2012

Messvektoren		Referenzvektoren	
Bezeichnung	Feldstärke in μT	Bezeichnung	Feldstärke in μT
M1	(0,245 14,738 -18,884)	P1	(-0,438 13,374 -19,174)
M8	(0,321 18,256 -28,869)	P2	(0,296; 17,496; -12,785)
M2-7	(0,406 19,013; -12,84)	P4	(0,55; 15,77; -17,935)
M1*	(-0,083 12,991 -20,163)	P5	(0,657; 16,675; -25,917)
M8*	(-0,128 19,957 -35,972)	P6	(0,114; 9,362; -17,302)
M2-7*	(0,076 18,386 -14,450)	P7	(-0,02; 22,171; -14,35)
		P8	(0,294; 18,591; -31,738)

Tabelle 3.2.: Ausgewählte Mess- und Referenzvektoren mit zugehörigen Magnetfeldstärken zur Verifikation der Tabelle 3.3 und 3.4

Messvektor	Ähnlichkeitsmaße					
		EA		KÄ		GÄ
M1	P4	1,4348 μT	P4	0,9982	P1	0,0099
	P1	1,5527 μT	P1	0,9981	P4	0,0098
M8	P8	2,8886 μT	P5	0,9998	P8	0,0018
	P5	3,3655 μT	P8	0,9994	P5	0,0010
M2-7	P2	1,5219 μT	P7	0,9996	P2	0,0102
	P7	3,5262 μT	P2	0,9993	P7	0,0015

Tabelle 3.3.: Die zwei ähnlichsten Referenzvektoren bei einer Ähnlichkeitssuche zu einem Messvektor. Die genutzten Abstandsmaße sind der euklidische Abstand (EA), Kosinus-Ähnlichkeit (KÄ) und die gaußsche Ähnlichkeit (GÄ). Die genutzten Messvektoren sind vom 8. November 2011.

Messvektor	Ähnlichkeitsmaße					
	EA		KÄ		GÄ	
M1*	P1	1,1188 μT	P5	0,9996	P1	0,0112
	P4	3,6174 μT	P1	0,9992	P4	0,0011
M8*	P8	4,4697 μT	P6	0,9999	P8	$4,4697 \times 10^{-4}$
	P5	10,607 μT	P8	0,9996	P5	$2,0249 \times 10^{-14}$
M2-7*	P2	1,9016 μT	P2	0,9993	P2	0,0065
	P7	3,7872 μT	P7	0,9957	P7	0,0013

Tabelle 3.4.: Die zwei ähnlichsten Referenzvektoren bei einer Ähnlichkeitssuche zu einem Messvektor. Die genutzten Abstandsmaße sind der euklidische Abstand (EA), Kosinus-Ähnlichkeit (KÄ) und die gaußsche Ähnlichkeit (GÄ). Die genutzten Messvektoren sind vom 16. November 2011.

Zum Vergleich sind in Tabelle 3.4 die Ergebnisse der Vektorähnlichkeit im selben Raum dargestellt. In diesem Fall wurden die Messvektoren zu einem späteren Zeitpunkt aufgenommen, um zu überprüfen, ob die initial aufgezeichneten Magnetfelder über längere Zeit stabil sind.

Diskussion

Die zuvor beschriebenen Messungen und Berechnungen dienten zur Darlegung, ob Magnetfelder für eine Ortsbestimmung in Gebäuden in Frage kommen. Dazu wurden für jeden Raum Messpunkte bei an- beziehungsweise ausgeschaltetem Strom aufgezeichnet. Diese Messungen wurden wiederholt, um zu überprüfen, ob die Magnetfelder in einem bestimmten Raum über längere Zeit stabil bleiben.

Die Tabelle 3.5 zeigt die Trefferquoten für die unterschiedlichen Suchanfragen. Bei der Kombinationsmessung wurden Messwerte bei angeschaltetem Strom mit den Referenzwerten bei ausgeschaltetem und umgekehrt verglichen. Mit der Kombinationsmessung sollte verglichen werden, ob Fingerprints die zum Beispiel bei ausgeschaltetem Strom aufgenommen wurden, auch weiterhin erkannt werden, wenn die Messung (Lokalisierung) bei angeschaltetem Strom erfolgt. Die Trefferquote lag bei den Messungen jeweils über 70%. Daraus wird geschlossen, dass die Wiedererkennung der Magnetfelder bei sich ändernden Bedingungen nach der Fingeraufnahme erfolgen kann.

Aus den Ergebnissen lässt sich ableiten, dass die Ortung mittels Magnetfeldern bei angeschalteten elektrischen Geräten in den getesteten Räumen besser funktioniert. Dies stützt die These, dass in Bürogebäuden das Erdmagnetfeld gestört ist und diese Störung zur Wiedererkennung von Messpunkten genutzt werden kann. Am besten schneidet in diesem Test die gaußsche Ähnlichkeit ab, am schlechtesten die Kosinus-Ähnlichkeit. Die gaußsche Ähnlichkeit sollte es ermöglichen, Messfehler die bei der Nutzung von Sensoren auftreten, auszugleichen. Das gering bessere Ergebnis verglichen mit dem euklidischen Abstand könnte darauf hindeuten. Da aber die Anzahl an Referenzpunkten pro Raum gering ist, kann dies nicht mit Sicherheit gesagt werden. In Abschnitt 6 werden die Vektormäße bei der Ortung in einer gesamten Etage untersucht.

Szenario	Trefferquote (in %)		
	euklidische Distanz	Kosinus-Ähnlichkeit	Gaußsche Ähnlichkeit
Strom an	81,48	70,37	81,48
Strom aus	46,66	60,00	46,66
Nachmessung Strom an	90,90	81,81	90,90
Nachmessung Strom aus	20,90	20,00	20,90
Kombinationsmessung	75,00	65,00	80,00
Gesamtergebnis	70,96	51,61	74,19

Tabelle 3.5.: Trefferquoten bei der Magnetfeldortung unter Berücksichtigung verschiedener Ähnlichkeitsmaße. Ein Treffer war, wenn für einen Messvektor der passende Referenzvektor unter den zwei ähnlichsten Ergebnissen war. Die Kombinationsmessung prüfte die Messwerte bei ausgeschaltetem Strom gegen die Referenzpunkte bei eingeschaltetem und umgekehrt. Die Werte für Strom an/aus enthalten auch die Ergebnisse der Nachmessung.

Die Räume *2.3.01* und *2.3.02* sind Büroräume mit mehreren Computerarbeitsplätzen, beide Messungen zeigten eine ähnliche Treffergenauigkeit (jeweils über 80 %). *2.2.1* hatte die geringste Erfolgsquote (zwei mal 50 % und 40 %), hingegen *2.2.18* die höchste (zwei mal 76 % und 80 %). Die genaueren Ergebnisse lassen sich durch die größere Anzahl an elektrischen Geräten erklären, die in dem Raum verteilt war. Ein weiterer Einflussfaktor könnte die geringere Fläche sein; die Störung durch metallische Gegenstände in den Wänden hätte demnach deutlicheren Einfluss auf das Feld.

Die Kombinationsmessung ergab trotz unterschiedlicher elektrischer Situationen in den Gebäuden eine hohe Wahrscheinlichkeit der korrekten Lokalisierung.

Auch bei der Nachmessung nach acht Tagen konnte bei angeschaltetem Strom eine hohe Trefferwahrscheinlichkeit erzielt werden. Die Quote bei ausgeschaltetem Strom hingegen war mit 20 % nicht ausreichend.

Bei den Berechnungen wurden nur die Referenzpunkte genutzt, die im selben Raum aufgenommen wurden. Das heißt, dass zuvor festgestellt werden musste, in welchen Raum die Ortung gerade stattfindet.

In einer weiteren Auswertung wurde untersucht, ob die Magnetfelder eindeutig zu unterscheiden sind, um bei der Lokalisierung alle im Gebäude gemessenen Referenzvektoren zu nutzen; eine Vorselektierung des Raumes wäre demnach überflüssig. Es zeigte sich, dass bei der Bestimmung des ähnlichsten Vektors keine nutzbaren Resultate erzielt werden konnten. Die Magnetfelder waren demnach zwar für einen Raum eindeutig, auf mehrere Räume gesehen jedoch nicht.

Die Ortung nur über den Einsatz der Ähnlichkeitsmaße eignet sich nicht für das spätere Szenario. Der Einsatz von Filtern, wie dem PF, ermöglichen es, bei einer Lokalisierung gleichfalls alte Ortungsergebnisse mit einzubeziehen. Wenn Magnetfelder zur Ortung im gesamten Gebäude genutzt werden sollen, muss ein Filter-Algorithmus genutzt werden.

3.1.3. Ortung über spezifische Lichtverhältnisse

Zusammen mit den Magnetfeldern wurden zudem die an den jeweiligen Punkten gemessene Lichtintensität aufgezeichnet. Dabei konnte festgestellt werden, dass beispielsweise bei Tageslicht zwischen Messpunkten am Fenster und von diesem entfernten unterschieden werden konnte. Gleiche Effekte traten ebenso bei künstlicher Beleuchtung auf. Messpunkte direkt unter Lampen unterschieden sich signifikant von weiter entfernten.

Grundsätzliche Eignung

Das größte Problem an Lichtfingerprints besteht in ihrer in der Regel nicht vorhandenen Kontinuität. Beispielsweise ist die Beleuchtungssituation zu unterschiedlichen Tageszeiten anders.

Besitzt das Gebäude, worin geortet wird, jedoch eine statische Lichtsituation (zum Beispiel in Museen mit wenigen Fenstern), kann der Lichtfingerprint benutzt werden, da kaum Schwankungen auftreten sollten.

Statt im Lichtfingerprint den tatsächlich gemessenen Wert zu speichern und somit dem Problem der Lichtschwankung zu unterliegen, könnten nur Lichtdifferenzen genutzt werden. Zum Beispiel könnte im Fingerprint verzeichnet werden, ob der Punkt „hell“ oder „dunkel“ ist. Dadurch kann unterschieden werden, ob die Messung an einem Fenster stattfand.

Die Messszenarien in dieser Arbeit wurden als statisch angesehen, folglich ist es ausreichend innerhalb der Fingerprints den tatsächlich gemessenen Wert zu speichern.

Ermittlung von Schwellwerten

Bei einer Ortung ohne statische Lichtverhältnisse gilt es zu ermitteln, ob sich der Lichtsensor in einem hellen oder dunklen Bereich befindet. Aufgrund der bereits genannten Lichtschwankungen kann kein fester Schwellwerte bei der Intensität angegeben werden.

Ein Ansatz wäre es, in dem Ortungssystem zusätzliche Lichtsensoren zu installieren. Diese könnten in den einzelnen Räumen verteilt werden um kontinuierlich die aktuelle Lichtintensität zu messen. Für den einzelnen Sensor müsste verzeichnet werden, ob er sich in einem hellen beziehungsweise dunklen Bereich befindet. Aus der Differenz der Lichtstärke am mobilen Gerät und der des Sensors im Raum, könnte geschlussfolgert werden in welchem Bereich das Gerät sich befindet.

3.1.4. Luftdruck zur Ermittlung der Etage

Wird die Ortung gleichfalls auf Etageebene durchgeführt, kann als Indikator der aktuelle Luftdruck genutzt werden. In den zuvor genannten Testreihen wurde dieser mit untersucht.

Es wurde festgestellt, dass der Luftdruck sich beim Wechseln der Etage verändert. Ähnlich der Lichtintensität ist dieser Messwert vom aktuellen Wetter abhängig, infolge dessen kann dieser

Wert nicht innerhalb des Fingerprints gespeichert werden, dennoch eignet sich der Luftdruck um innerhalb eines kurzen Zeitraumes Differenzen zu erkennen [Str+08, S. 2].

Aus der Differenz zwischen zwei Luftdruckwerten kann die Veränderung der Etage hergeleitet werden. Sinkt der Luftdruck, wird eine höher gelegene Etage benutzt und umgekehrt. Dieses Wissen kann genutzt werden, um den Suchraum der Fingerprints zu reduzieren. Statt die Fingerprints des gesamten Gebäudes zu betrachten, brauchen nur noch die der aktuellen Etage in Betracht gezogen zu werden.

In Abschnitt 3.2.5 wird beschrieben, wie diese Informationen in der Ortungsstrategie genutzt werden.

3.2. Partikel-Filter zur Zusammenführung von Sensordaten

Magnetfelder haben sich in den beschriebenen Tests als grundsätzlich tauglich für die Ortung in Gebäuden erwiesen. Die Nutzung von Filter-Algorithmen erlauben es, Positionsschätzungen zu verbessern, indem alte Schätzungen mit berücksichtigt werden. In diesem Abschnitt wird die konkrete Umsetzung des PF für das Zusammenführen von Sensordaten und der daraus resultierenden Ortung beschrieben. Zunächst wird erläutert, warum sich der PF eignet um die genutzten Sensortypen zu vereinen.

3.2.1. Auswahl des Filter-Algorithmus

Die in der Literatur meist genutzten Filter-Algorithmen sind zum einen der KF und PF. Während der KF zunächst nur für lineare Systeme geeignet ist, gibt es Erweiterungen für nicht lineare Systeme [siehe JU97]. Selbst die nicht linearen Erweiterungen lassen es jedoch nicht zu, dass der geschätzte Systemzustand nicht eindeutig ist.

Als Beispiel dient eine reine Ortung über Lichtfingerprints. Wie zuvor genannt, sind diese meist nicht eindeutig. Wird die Lichtstärke entlang einer Fensterfront gemessen, ist diese an mehreren Punkten mindestens sehr ähnlich. Soll in einer späteren Lokalisierungsphase die Position über die Lichtstärke ermittelt werden, kann höchstens die Aussage getroffen werden, dass die Messung *irgendwo* am Fenster statt fand.

Würde die Filterung über eine Variation des KF erfolgen, müsste algorithmusbedingt die Position auf einen der Fingerprints geschätzt werden. Die Information, dass das zu ortende Objekt sich prinzipiell irgendwo an der Fensterfront befinden könnte, geht verloren. Der PF lässt eine solche Modellierung hingegen zu. Die Partikel würden sich in diesem Fall auf die Fingerprints entlang der Front verteilen und dort bleiben, da die Wahrscheinlichkeit sich irgendwo zu befinden gegeben ist. Erst bei besseren Signalen, etwa über eine beliebige Funklateration, könnte die Position präzisiert werden.

Aufgrund der verwendeten Ortungsansätze und der damit verbundenen Vorteilen für den PF wird in dieser Arbeit selbiger genutzt.

Die Literatur ist über die Eignung der Filter-Algorithmen im Gebäudeortungsszenario nicht eindeutig. Es sei auf [YM02; EM06; Wan+07b; Gus+02] verwiesen, die den Einsatz des PF und [Mih+06; Au10], welche den KF aufgrund der besseren Ressourceneffizienz, bevorzugen.

3.2.2. Zustandsraum

Der PF schätzt die Wahrscheinlichkeit, dass sich ein System in einem bestimmten Zustand befindet. Dieser allgemeine Fall muss für den Anwendungsfall der Ortung konkretisiert werden. In dieser Arbeit wird eine Lokalisierung innerhalb eines Gebäudes entworfen. Der Zustandsraum besteht somit aus der zu ortenden Fläche und der jeweiligen Etage (siehe Abschnitt 2.5.1). Der Zustandsvektor um diese Position zu repräsentieren ist:

$$x = \begin{pmatrix} p^x \\ p^y \\ p^z \end{pmatrix} \quad (3.1)$$

Der Abstand der Zustände ergibt sich zunächst durch die Granularität der Fingerprintkarte. Das bedeutet bei einer Ortung, dass bei einem Messfehler mindestens eine Abweichung vom Abstand zwischen den Fingerprints entsteht. Eine Lösung zur Erhöhung der Fingerprintdichte ist es entweder, diese in kürzeren Abständen aufzunehmen oder Zwischenmesswerte durch Interpolation zu berechnen.

3.2.3. Genutzter Messvektor

Eine neue Iteration bei einem PF wird durchgeführt, wenn neue Messwerte zur Verfügung stehen (siehe Abschnitt 2.5.5). Dieser Messvektor wird innerhalb des Messmodells verarbeitet. Das in dieser Arbeit entwickelte Konzept zur Lokalisierung benutzt Messwerte aus dem Magnetfeld- und Lichtsensor sowie dem Barometer. Diese Daten werden zusammen in einem jeweiligen Fingerprint gespeichert, deshalb werden sie in der Lokalisierung zusammen ausgewertet. Der Messvektor wird somit durch

$$z = \left(m_x \quad m_y \quad m_z \quad l \quad a \right)^T \quad (3.2)$$

beschrieben. Der genutzte Magnetfeldsensor ermittelt das Feld in alle drei Freiheitsgrade, folglich sind m die normalisierten Komponenten des Magnetfeldes, l die Lichtintensität und a der aktuelle Luftdruck.

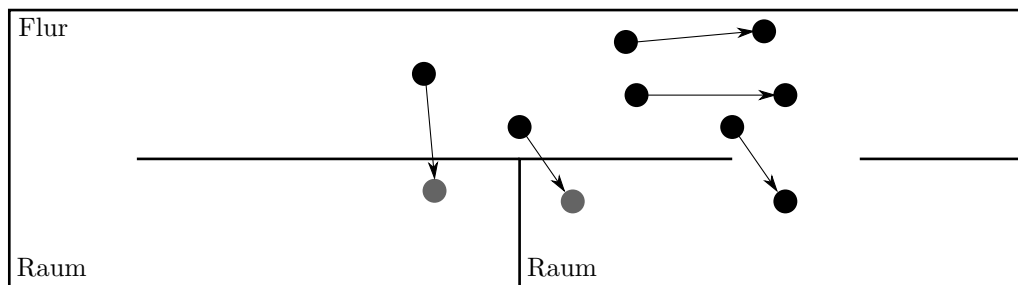


Abbildung 3.2.: Schematische Darstellung einer Partikelbewegung mit anschließender Bewertung beim Durchqueren einer Wand. Graue Partikel haben dabei eine Wand durchquert.

3.2.4. Repräsentation einer Karte

Zur Verbesserung der Genauigkeit wird die unter anderem in Widyawan [Wid10] vorgestellte Erweiterung des Messmodells genutzt. Dabei werden vorhandene Karteninformationen verwendet, um die Bewegung der Partikel zu bewerten. Bewegt sich ein Partikel während des Bewegungsmodells durch eine Wand, wird das Gewicht auf 0 gesetzt.

In Abbildung 3.2 ist eine Partikelbewegung auf einer Karte dargestellt. Die grau eingefärbten Partikel haben aufgrund des Bewegungsmodells eine Wand durchquert.

Der Grundriss eines Gebäudes wird durch eine Ansammlung von Wänden bestimmt. Im Messmodell wird anschließend für jeden Partikel bestimmt, ob eine Wand durchkreuzt wurde.

3.2.5. Bewegungsmodell

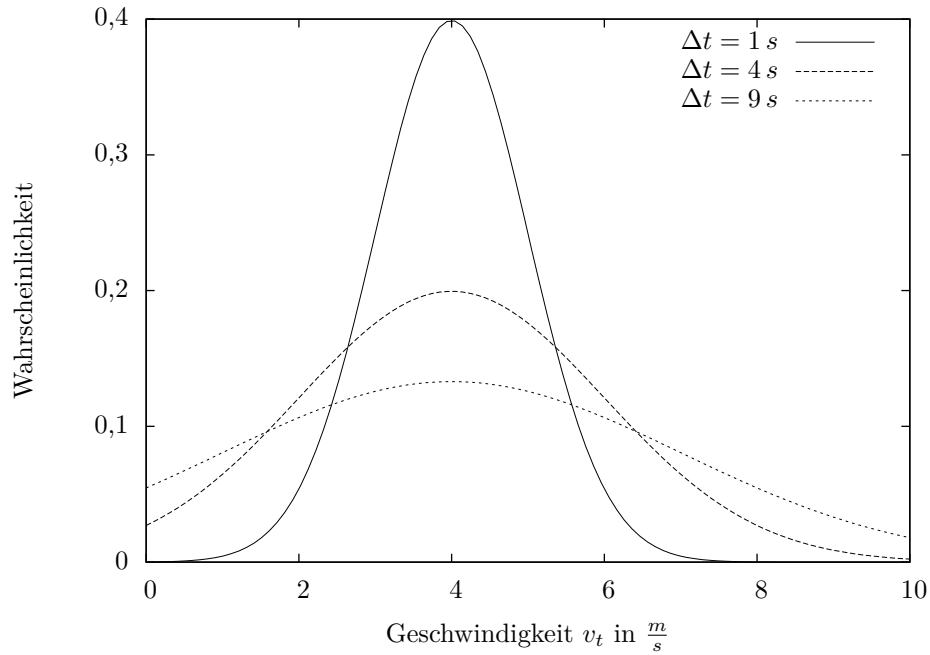
Das Bewegungsmodell soll beschreiben, wie sich das zu ortende Gerät im Gebäude bewegt. Für gewöhnlich wird das Gerät von einer Person getragen, dadurch kann die Bewegung einer Person modelliert werden.

Geschwindigkeit des Partikels

In Abschnitt 2.5.4 wurde die allgemeine Bewegung im genutzten Zustandsraum beschrieben. Es gilt nun, die Parameter der Fortbewegung zu bestimmen. Das in dieser Arbeit genutzte Modell wird an das in Widyawan [Wid10] beschriebene angelehnt. Die dafür genutzten Parameter wurden in Bruce und Gordon [BG04] experimentell bestimmt und anschließend in das Modell überführt.

Dieses sieht vor, dass die Bewegung eines Partikels eine zufällige Komponente, basierend auf dem alten Zustand, ist.

Die maximale Geschwindigkeit und somit die gelaufene Strecke, besitzt für Menschen Grenzen. Beispielsweise kann eine Person nur eine bestimmte Geschwindigkeit erreichen und diese wird sie meist nicht abrupt verändern [Wid10, S. 34]. Die neue Geschwindigkeit v für einen Partikel wird für eine Iteration wie folgt beschrieben:


 Abbildung 3.3.: Gaußverteilungen für das Bewegungsmodell bei $v_{t-1} = 4 \text{ m s}^{-1}$ [nach Wid10, S. 34]

$$v_t = \mathcal{N}(v_{t-1}, \sigma_v), v_t = \begin{cases} v_t, & 0 \leq v_t \leq \text{Max}_{v_t} \\ |v_t|, & v_t < 0 \\ \text{Max}_{v_t}, & v_t > \text{Max}_{v_t} \end{cases} \quad (3.3)$$

$$\sigma_v = \min(\text{Max}_{\Delta t}, \sqrt{\Delta t}) \quad (3.4)$$

\mathcal{N} beschreibt die Ziehung einer gaußverteilten Zufallsvariable, mit dem Erwartungswert v_{t-1} als zuletzt ermittelte Geschwindigkeit und σ_v als Standardabweichung. Die Abweichung berücksichtigt die seit dem letzten Messwert vergangene Zeit und die damit zurückgelegte Strecke. Max_{v_t} ist die maximale Geschwindigkeit, die die Person zurücklegen kann und wurde in dieser Arbeit auf 10 m s^{-1} und $\text{Max}_{\Delta t}$ auf 3 s festgelegt [Wid10, S. 34]. Diese Funktion hängt von der zeitlichen Differenz zwischen zwei Messwerten ab. Ist diese Differenz hoch, dann kann nach diesem Modell die neue Geschwindigkeit der Person eher sich verändert haben, als wenn die Messwerte in kurzen Intervallen eintreffen. Die Relation zwischen neuer Geschwindigkeit und der Zeitdifferenz zwischen zwei Signalen, wird in Abbildung 3.3 dargestellt.

Richtung des Partikels

Neben der Geschwindigkeit muss auch ein Wert für die Richtung der Bewegung ermittelt werden. Diese wird über

$$\alpha_t = \mathcal{N}(\alpha_{t-1}, \sigma_\alpha) \quad (3.5)$$

$$\sigma_\alpha = 0,4\pi - \arctan\left(\frac{\sqrt{v_{t-1}}}{2}\right) \quad (3.6)$$

festgelegt. Der Erwartungswert ist erneut die alte Bewegungsrichtung, da es unwahrscheinlicher ist, dass diese plötzlich gewechselt wird. Die Standardabweichung dient dazu, bei hohen Geschwindigkeit die Richtungsänderung klein zu halten. Die Annahme ist, dass bei hoher Geschwindigkeit eher eine gerade Linie gegangen wird, während beim Umgehen einer Ecke die Geschwindigkeit verringert wird [Wid10, S. 35].

Mit Formel 2.22 wird der Partikel anschließend in die zuvor ermittelte Richtung verschoben.

Auswahl der Etage

Der Luftdruck als Etagenindikator liefert keine unmittelbaren Positionsinformationen stattdessen wird es im Bewegungsmodell verwendet, um die Partikel innerhalb der Etagen zu verschieben. Durch die Funktion

$$\Psi(a_t, a_{t-1}) = \begin{cases} 1, & a_t < a_{t-1} \\ -1, & a_t > a_{t-1} \\ 0, & a_t \approx a_{t-1} \end{cases} \quad (3.7)$$

wird der Übergang innerhalb der Etagen aufgrund der Differenz des letzten und aktuellen Luftdrucks bestimmt. Der mittels Barometer gemessene Luftdruck, gibt die Luftsäule über dem Sensor an. Wenn der Sensor sich höher befindet, wird gleichermaßen die Luftsäule kleiner und somit der Luftdruck.⁴ Zur Ermittlung des Etagenwechsels, muss kontinuierlich der aktuelle Luftdruck mit dem zuletzt gemessenen verglichen werden. Ist der aktuelle Wert signifikant kleiner als der zuvor gemessene, kann davon ausgegangen werden das das mobile Gerät eine Etage höher bewegt wurde.

In diesem Schritt können ebenfalls Informationen aus der Umgebungskarte genutzt werden. Befindet der Partikel sich nicht in der Nähe eines Treppenhauses (oder ähnliches) ist es unwahrscheinlich, dass die Etage gewechselt wurde.

⁴vgl. <http://wetter.mb.fh-jena.de/station/klima/luftdruck.html> Abruf 16. Februar 2012

3.2.6. Messmodell

Fingerprints für Magnetfeld- und Lichtdaten

Eine Möglichkeit der Berechnung von Partikelgewichten ist es, die Distanz zu einem Emmitter zu bestimmen. Zum Beispiel wird dies beim RSSI Verfahren bei W-LAN genutzt. In diesen Fall könnte der Zustandsraum beliebig klein gewählt werden, da für jeden Punkt ein mehr oder weniger eindeutiger RSSI ermittelt werden kann. Voraussetzung dafür ist, dass ein *Propagation Model* für die Ausbreitung der Funkwellen verfügbar ist.

In dieser Arbeit sollen zur Lokalisierung unter anderem eindeutige Magnetfelder genutzt werden. Diese werden vor der eigentlichen Ortung aufgenommen und in eine Datenbank zusammen mit ihrer Position gespeichert. Ein Propagation Model kann für diesen Fall kaum ermittelt werden, da in diesem alle das Magnetfeld beeinflussende Objekte berücksichtigt werden müssten. In einem Gebäude, in dem Stahlträger in den Wänden das Magnetfeld ändern können, ist dies nicht zu realisieren. Infolge dessen wurde sich für einen Fingerprintansatz entschieden, bei dem das Feld empirisch an vordefinierten Stellen aufgenommen wird.

In Abschnitt 3.1.2 wurde beschrieben, dass die Drehmatrix des Tablets nicht korrekt ist, welche sich auf die normalisierten Magnetfelder auswirkt. Wird etwa das Tablet auf der selben Stelle gedreht, ändert sich der Magnetfeldvektor, obwohl dieser (wegen der Normalisierung) gleich bleiben sollte.

Ein Lösungsansatz ist es, für jede Himmelsrichtung einen eigenen Messvektor aufzuzeichnen. Diese Richtung kann zum Beispiel über den Magnetfeldsensor bestimmt werden, weil der aufgezeichnete Vektor in Richtung magnetischen Norden zeigt. Ein Fingerprint besteht aus dem Zustand und vier Messvektoren, einen für jede Himmelsrichtung:

$$F_x = \langle x, \{ \langle h, z \rangle \mid h \in H \} \rangle \quad (3.8)$$

$$H = \{ N, S, W, O \} \quad (3.9)$$

Die spätere Ortungsgenauigkeit hängt von dem Abstand der Fingerprints zueinander ab. Um den Messaufwand zunächst gering zu halten, wurden in einem Büroraum mit zirka 28 m² fünf Fingerprints aufgenommen. Für diesen werden entsprechend die drei Komponenten des gemessenen Magnetfeldes und die aktuelle Lichtintensität abgespeichert. Abbildung 3.4 visualisiert das Magnetfeld für einen Büroraum. Der gespeicherte Fingerprint entspricht dem in Abschnitt 3.2.3 eingeführten Messvektor.

Zusammenführen von Sensordaten

Das Messmodell dient dazu Daten verschiedener Sensoren miteinander zu korrelieren und zu bewerten wie ähnlich Fingerprintvektor und aktueller Messvektor sind. In Abschnitt 2.2 wurden

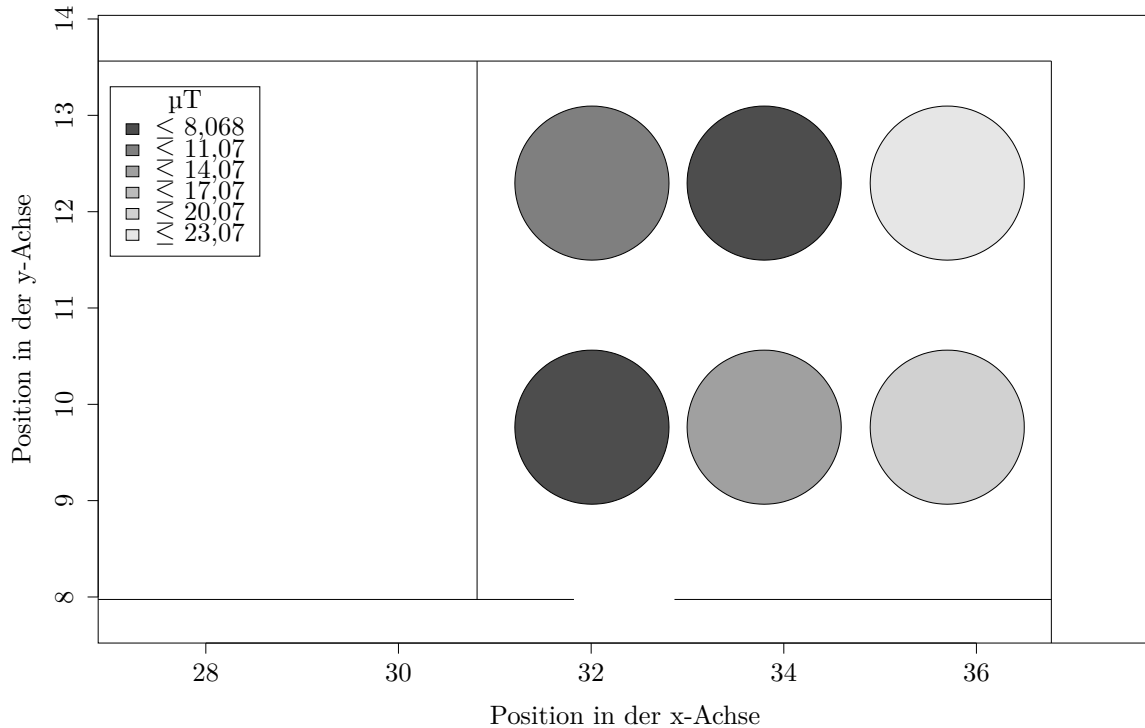


Abbildung 3.4.: Magnetfeldverteilung im Büroraum 2.3.02; am Beispiel der x-Achse

verschiedene Möglichkeiten der Ähnlichkeitssuche beschrieben. Der Abschnitt 3.1 führt zudem die Ortungsergebnisse mit verschiedenen Ähnlichkeitsmaßen auf.

Messungen des Magnetfeldes an einer Stelle haben ergeben, dass die Sensordaten zum Teil stark streuen. In Referenz auf [Au10; Wid10] wurde sich dementsprechend für eine wahrrscheinlichkeitsbasierte Ähnlichkeitssuche entschieden. Die Gesamtwahrscheinlichkeit im Messmodell $P(z_t | x_t)$ ergibt sich aus dem Produkt der Teilähnlichkeiten der einzelnen Dimensionen. Dies wird durch folgende Formel beschrieben:

$$P(z_t | x_t) = P(m_{x,t} | x_t) \cdot P(m_{y,t} | x_t) \cdot P(m_{z,t} | x_t) \cdot P(l_t | x_t) \quad (3.10)$$

Die Ähnlichkeit einer einzelnen Dimension wird über die gaußsche Verteilung bestimmt. Im Folgenden ist sie für die x-Dimension des Magnetfeldes dargestellt und wird auf die anderen Dimensionen des Messvektors genauso angewandt.

$$P(m_{x,t} | x_t) = \frac{1}{\sigma_{m_x} \sqrt{2\pi}} \exp\left(-\frac{(m_{x,t} - m_{x,t}^*)^2}{2\sigma_+^2}\right) \quad (3.11)$$

Wobei σ_{m_x} das ermittelte Messrauschen der x-Ebene des Magnetfeldes und $m_{x,t}^*$ der Magnetfeldwert in der x-Ebene des Fingerprints des Zustandes x_t ist.

Eine weitere Möglichkeit der Ähnlichkeitsbestimmung ist der Nächste Nachbar Ansatz. Dabei werden gemäß Abschnitt 2.2 die Fingerprints über euklidische Distanzen verglichen. Das Ergebnis ist eine Reihenfolge, welche Fingerprints der Messung am ähnlichsten sind. Um aus dieser Reihenfolge eine Wahrscheinlichkeit abzuleiten, wird die Formel

$$P(z_t | x_t) = 0,5^{i_t} \quad (3.12)$$

genutzt. Wobei i_t die Position des Zustands x_t in einer aufsteigend sortierten Liste ist, welche nach der euklidischen Distanz zum Vektor z_t sortiert wurde [GGH10]. Diese Berechnung hat zur Folge, dass umso weiter entfernt der Fingerprint im Vergleich zur Messung ist, umso geringer ist die errechnete Wahrscheinlichkeit. Die gleiche Art der Berechnung wird ebenso bei der Kosinus-Ähnlichkeit verwendet. Es geht bei dieser Berechnung nur um die Reihenfolge der Fingerprints verglichen zu dem empfangenen Messwert.

Die Messdaten des Luftdrucks werden hingegen nicht im Messmodell verwendet. Sie dienen dazu, die Partikel im Bewegungsmodell entsprechend zu verschieben. Das Messmodell hingegen bewertet, wie das Bewegungsmodell korrekt geschätzt hat. Eine Bewertung der Partikelbewegung ist demnach nicht möglich, da diese bereits auf den Messwerten des Luftdrucks basiert.

Bestimmung der Modellparameter

Wie zuvor beschrieben, müssen für die Fingerprintsuche mit der gaußschen Ähnlichkeitsfunktion die Standardabweichung festgelegt werden. Zur Bestimmung dieser für den Magnetfeldsensor wurde an einer festen Stelle das Feld aufgenommen. Aus 2000 aufgenommenen Messwerten wurde für jede Achse des Sensors eine gaußsche Dichtefunktion approximiert. Diese ist in Abbildung 3.5 dargestellt. Diese Funktion besitzt für x-Achse eine Standardabweichung von zirka $0,64 \mu\text{T}$, für y und z jeweils eine von $0,53 \mu\text{T}$.

3.3. Ablauf einer Ortung

In den Abschnitten zuvor wurden die einzelnen Komponenten der magnetfeld- und lichtbasierten Ortung beschrieben. Nachfolgend wird der gesamte Ortungsprozess beschrieben (siehe Abbildung 3.6).

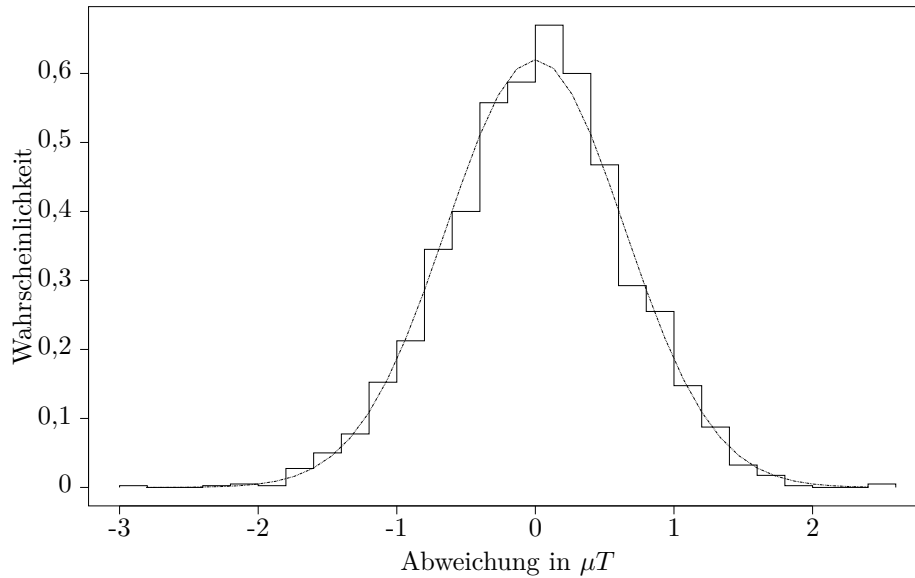


Abbildung 3.5.: Ermittelte Modellparameter für das Bewegungsmodell. Abgebildet sind Messfehler des Magnetfeldes in der x Ebene

3.3.1. Offline-Phase

In der Offline-Phase werden für die Fingerprints die Messvektoren aufgenommen. Wie zuvor beschrieben, wird wegen der besseren Genauigkeit, für jede Himmelsrichtung ein Messvektor abgespeichert. Aufgrund der ermittelten Messfehler (siehe Abschnitt 3.2.6) werden mehrere Sensorwerte gelesen und aus diesen ein Mittelwert gebildet. In dieser Arbeit wurden sieben Messwerte genutzt, um daraus den Messvektor zu bilden.

Die nach Formel 3.8 aufgebauten Fingerprints werden anschließend in einer Datenbank auf dem mobilen Endgerät gespeichert.

3.3.2. Online-Phase

Die Online-Phase ist der eigentliche Ortungsprozess, wobei eine Fingerprintdatenbank die Voraussetzung ist. Zunächst wird der PF initialisiert, in dem alle Partikel im Zustandsraum gleich verteilt werden.

Für jeden neuen Messwert wird eine Partikel-Filter-Iteration durchlaufen, bestehend aus dem Bewegungsmodell und anschließend dem Messmodell. Die Himmelsrichtung h dient dazu, den korrekten Messvektor aus der Fingerprintdatenbank für den Filter im Messmodell zu ermitteln.

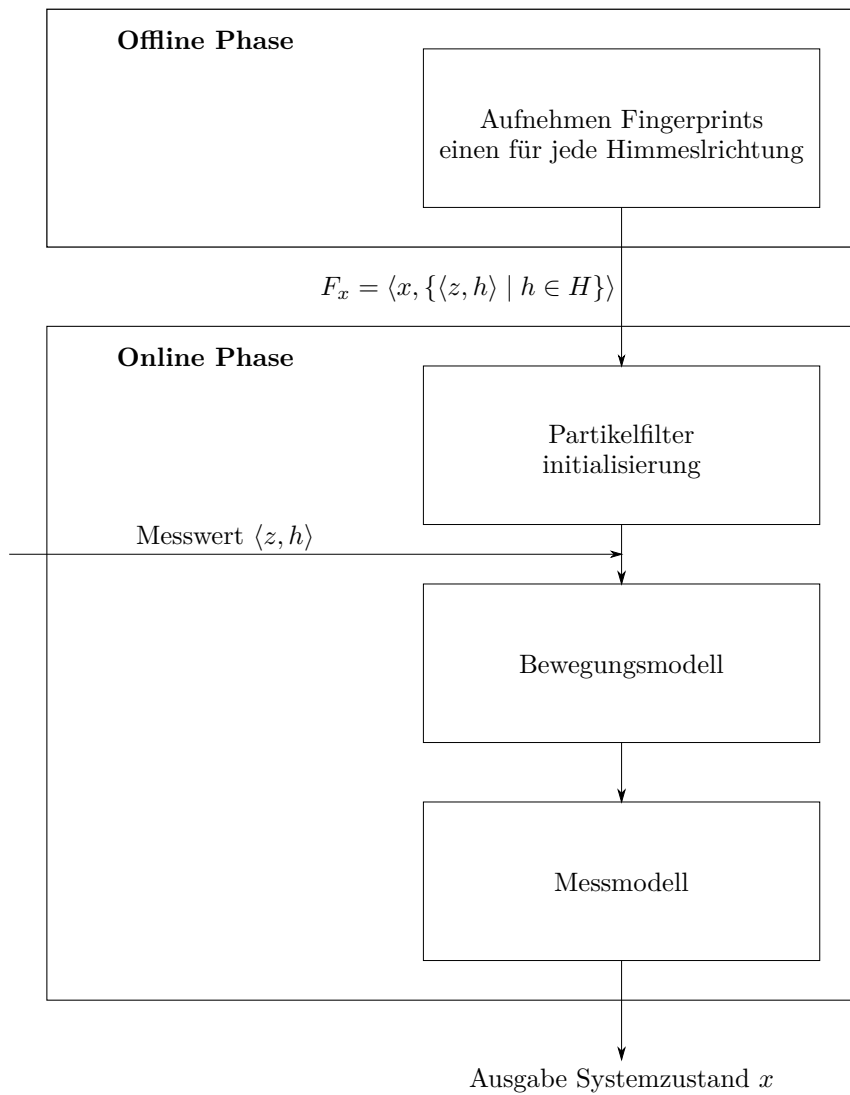


Abbildung 3.6.: Ablaufplan der Ortung mittels Partikel-Filter

4. Navigation in Gebäuden

Die Navigation dient dem Nutzer in unbekanntem Gebäuden das gewünschte Ziel zu erreichen. In diesem Abschnitt wird beschrieben, wie eine Navigation auf dem mobilen Gerät erfolgen kann. Dabei gilt zu berücksichtigen, dass es vor allem beim Etagenwechseln verschiedene Möglichkeiten gibt. Die unterschiedliche Geschwindigkeit bei diesen Wegen muss in dem Navigationsalgorithmus beachtet werden. Weiterhin wird ein Weg vorgestellt, wie aus den Fingerprints, welche für die Ortung genutzt werden, auch ein Graph für den Algorithmus erstellt werden kann. Neben den technischen Voraussetzungen für die Ortung müssen Wege geschaffen werden dem Nutzer den errechneten Pfad zu visualisieren.

4.1. Kartenrepräsentation

Um einen Routingalgorithmus wie Dijkstra anwenden zu können, muss ein Graph mit einer entsprechenden Kostenfunktion für die Kanten vorliegen. Im Folgenden Abschnitt wird ein Algorithmus beschrieben, der einen Graphen aus den für die Ortung genutzten Fingerprints ableitet.

Der letzte Abschnitt beschreibt, wie eine Kostenfunktion für Kanten im Gebäude erstellt werden kann. Dabei sind zwei Fälle zu unterscheiden: zwei Knoten befinden sich in derselben bzw. in einer unterschiedlichen Etage.

4.1.1. Nutzung der Fingerprints

Im vorherigen Kapitel wurde eine Ortungsstrategie basierend auf Fingerprinttechniken vorgestellt. Die dort genutzten Fingerprints, welche neben den Messdaten gleichfalls die Position derer festhalten, können für die Navigationslösung verwendet werden.

Weiterhin sind bei der Ortung bereits Daten zum Grundriss des Gebäudes aufgenommen worden (siehe Abschnitt 3.2.4). Diese Daten können genutzt werden, um die Kanten unter den Knoten automatisch aufzubauen. Die Fingerprints liegen in einem regelmäßigen Raster vor, sodass jeder vier direkte Nachbarn hat. Der Algorithmus 4.1 wird genutzt, um aus den Fingerprints einen Graph zu erstellen.

Ausgangspunkt ist die Menge der Fingerprints F und Wände W . Für jeden Fingerprint werden die vier Nachbarn ermittelt und darauf geprüft, ob die Strecke zwischen ihnen eine Wand schneidet. Nach diesem Algorithmus sind alle Knoten in einem Raum und die Übergänge zwischen den

Algorithmus 4.1 Aufbau eines Graphens aus Fingerprints und Wänden (F, W)

```

1:  $V \leftarrow \emptyset$  ▷ Knoten
2:  $E \leftarrow \emptyset$  ▷ Kanten
3: for all  $f \in F$  do
4:    $V \leftarrow V \cup \{f\}$ 
5:    $N \leftarrow$  Ermittle-Nachbarn ( $f$ )
6:   for all  $n \in N$  do
7:      $schneidet \leftarrow$  false
8:     for all  $w \in W$  do
9:       if Schneiden ( $\overline{fn}, w$ ) then
10:         $schneidet \leftarrow$  true
11:        break
12:      end if
13:    end for
14:    if  $\neg schneidet$  then
15:       $E \leftarrow E \cup \{\overline{fn}\}$ 
16:    end if
17:  end for
18: end for

```

Türen verbunden. Ergebnis ist ein Graph $G = (V, E)$, welcher das Gebäude repräsentiert. Die Abbildung 4.1 stellt den Zusammenhang zwischen den Fingerprints und den daraus resultierenden Kanten dar.

4.1.2. Semantischer Zusammenhang der Daten

Alle Fingerprints, die zu einer semantischen Einheit gehören, werden mit zusätzlichen Informationen angereichert. Es werden Knoten mit einem Namen pro Raum versehen, der es ermöglicht Navigationsziele spezifischer auszugeben.

Knoten, die zum Typ „Treppe“ beziehungsweise „Fahrstuhl“ gehören, werden mit dem entsprechenden markiert, damit das Navigationsmodul für diese Punkte spezielle Richtungsanweisungen geben kann.

Wird bei einer Navigation als Ziel ein Raum angegeben, wird als Zielknoten der Knoten ausgewählt, welcher sich am Übergang zum Flur oder einem anderen Raum befindet.

4.1.3. Kostenfunktion zwischen den Knoten

Die Kostenfunktion für eine Kante zwischen zwei Knoten ist essenziell für den Routingalgorithmus, da auf ihm basierend die Suche des besten Pfades durchgeführt wird. Basis der Kostenfunktion ist in dieser Arbeit die maximale Geschwindigkeit, die über eine jeweilige Kante erreicht werden kann. Verglichen mit Fahrzeugnavigationssystemen ergeben sich in Gebäuden ähnliche Überlegungen zu den Kantengewichten. Straßen besitzen zum Beispiel unterschiedliche Höchstgeschwindigkeiten; so

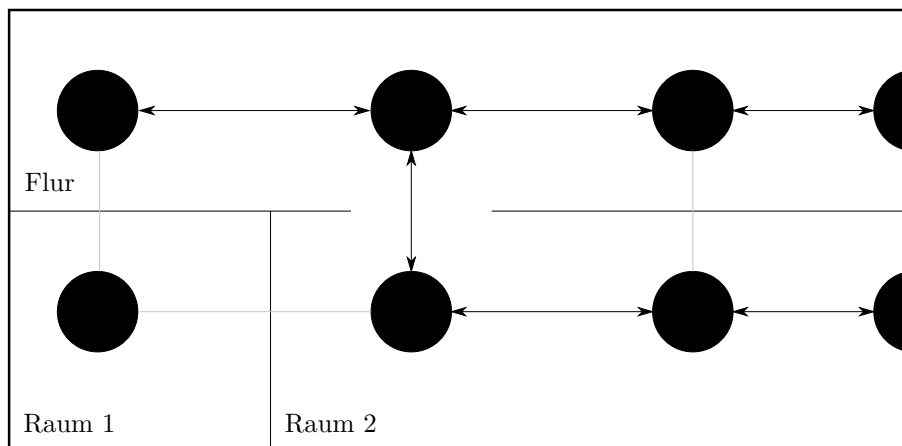


Abbildung 4.1.: Schematische Darstellung der Kanten nach der Erstellung des Graphens aus den Fingerprints in unterschiedlichen Räumen. Die schwarzen Kanten sind die berechneten Kanten. Die grauen Linien zeigen die Nachbarbeziehung zu anderen Fingerprints, welche allerdings eine Wand schneiden und damit ungültig sind.

ist es oft von Vorteil die Autobahn für längere Strecken zu nutzen, da auf ihnen die maximale Geschwindigkeit am höchsten ist.

Die Geschwindigkeit in Gebäuden hängt somit ebenfalls von der Art des genommenen Weges ab. Zwischen Räumen auf einer Etage erfolgt die Fortbewegung zum Beispiel über Flure. Muss die Etage gewechselt werden, stehen üblicherweise Treppen oder Fahrstühle zur Verfügung.

Die Laufgeschwindigkeit in der Ebene und besonders beim Treppengehen hängt von der individuellen Fähigkeit des Benutzers ab. Ebenso ist die Geschwindigkeit bei der Benutzung des Fahrstuhls vom Fahrstuhltyp und Faktoren wie der Wartezeit auf selbigen bestimmt. Zur Vereinfachung der Kostenfunktion werden in dieser Arbeit von den jeweiligen Faktoren abstrahierte Geschwindigkeiten genutzt.

Bei der Bewegung in der Ebene wurde als Geschwindigkeit $1,19 \text{ m s}^{-1}$, beim Treppengehen $0,24 \text{ m s}^{-1}$ und bei der Nutzung eines Fahrstuhls $0,188 \text{ m s}^{-1}$ festgesetzt. Diese Geschwindigkeiten wurden empirisch im Testgebäude bestimmt und sollen einen Eindruck darüber vermitteln, wie die Benutzung von unterschiedlichen Pfaden im Gebäude eine andere Geschwindigkeit erbringt.

Die folgende Gleichung stellt die Kostenfunktion zwischen zwei Knoten im Graph dar. Die x- und y-Koordinate sind die Positionen der Knoten auf dieser Ebene in Metern, die z-Koordinate beschreibt die Etage. Um die Distanz für z zu ermitteln, muss die Differenz der Etage mit der Höhe einer Etage h multipliziert werden.

Die Kostenfunktion basiert auf der euklidischen Distanz zweier Knoten multipliziert mit einem Kostenfaktor für die jeweilige Kante. Diese Funktion wird durch die folgende Gleichung beschrieben:

$$d(a, b) = \sqrt{(a_x - b_x)^2 + (a_y - b_y)^2 + ((a_z - b_z) \cdot h)^2} \cdot v \quad (4.1)$$

$$v = \begin{cases} 1,19 \frac{m}{s}, & a_z = b_z \\ 0,24 \frac{m}{s}, & Treppe \\ 0,188 \frac{m}{s}, & Fahrstuhl \end{cases} \quad (4.2)$$

4.2. Ablauf der Navigation

Ziel der Navigation ist es den Weg von einem Start- zu einem Endpunkt zu berechnen und den Benutzer anschließend mit Richtungsanweisungen entlang des Pfades zu führen. In Abbildung 4.2, angelehnt an [Au10, S. 60], ist der allgemeine Ablauf einer Navigation dargestellt. In den folgenden Abschnitten werden die einzelnen Schritte näher erläutert.

4.2.1. Beginn des Routings

Am Anfang der Navigation steht der Wunsch des Nutzers zu einem bestimmten Ziel zu gelangen. Dieses Ziel kann entweder durch direkte Eingabe erfolgen, wie zum Beispiel „Raum 2.3.01“ oder durch Auswahl des Ziels auf einer Übersichtskarte.

Mittels des ausgewählten Ziels und der durch das Ortungsmodul errechneten Position als Start kann anschließend eine Initialroute errechnet werden. Beide Punkte sind in diesem Szenario Fingerprints in dem zuvor errechneten Graphen.

4.2.2. Pfaderstellung

Die Pfaderstellung erfolgt durch den Dijkstra-Algorithmus (siehe Abschnitt 2.6.2). Abschnitt 4.1.1 beschreibt die Berechnung des Graphen. Voraussetzung ist, dass der Graph zusammenhängend ist, damit von jedem Startpunkt das Ziel durch eine Abfolge von Kanten erreicht werden kann [Au10, S. 61]. Der Algorithmus wird dahingehend angepasst, dass die Berechnung der kürzesten Pfade abgebrochen wird, falls der Zielknoten in die Liste der bereits besuchten Knoten aufgenommen wurde.

Eine Ausgabe des Dijkstra-Algorithmus ist die Liste π , welche für einen Knoten den Vorgänger angibt. Mit Algorithmus 4.2 wird aus dieser Liste die Routingpunkte erstellt.

4.2.3. Periodische Positionsbestimmung

Um die Navigation an die derzeitige Position des Nutzers anzupassen, ist es notwendig seine errechnete Position in das Navigationsmodul einfließen zu lassen. Die Navigation ist somit eng mit der

Algorithmus 4.2 Erstellen des Routing-Pfades (π, s, z)

```

1:  $R \leftarrow \{z\}$ 
2:  $t \leftarrow z$ 
3: while  $t \neq s$  do
4:    $t \leftarrow \pi[t]$ 
5:    $R \leftarrow \{t\} \cup R$ 
6: end while

```

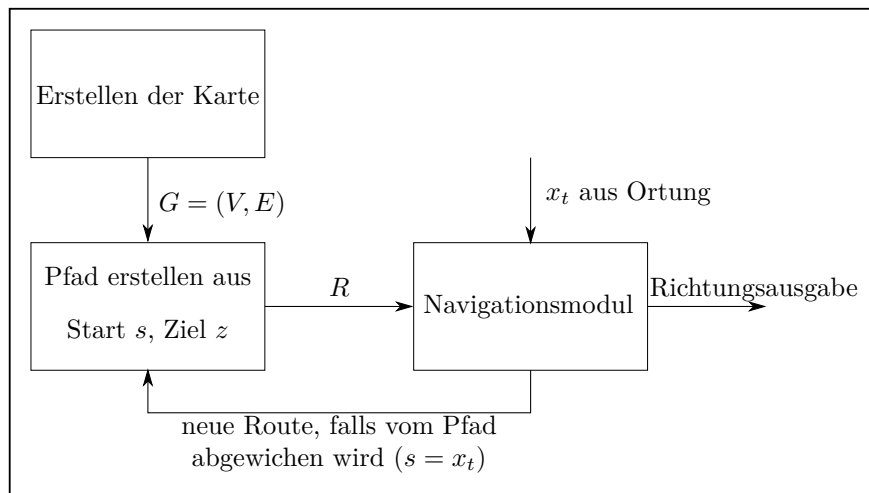


Abbildung 4.2.: Ablauf einer Navigation

Ortung verbunden und die Qualität der Navigationsanweisungen hängt auch von der Genauigkeit der Positionsermittlung ab.

Es ist vorgesehen, dass für jeden Messwert und somit für jede neu errechnete Position, das Navigationsmodul über diese informiert wird.

4.2.4. Navigationsmodul

Neuberechnung der Route

Die Aufgabe des Navigationsmodul ist es, zu verfolgen, ob der Nutzer entsprechend der berechneten Route durch das Gebäude läuft. Weiterhin sollen an markanten Navigationspunkten, wie Kreuzungen, dem Nutzer Anweisungen zur weiteren Route gegeben werden.

Die Eingabe für dieses Modul ist der Pfad R , welcher die Knoten auf dem Weg zum Ziel enthält. Wie zuvor erwähnt, wird periodisch die aktuelle Position aus dem Ortungsmodul an die Navigation übertragen. Für jede Position wird überprüft, ob

$$R \setminus \{x_t\} = \emptyset \tag{4.3}$$

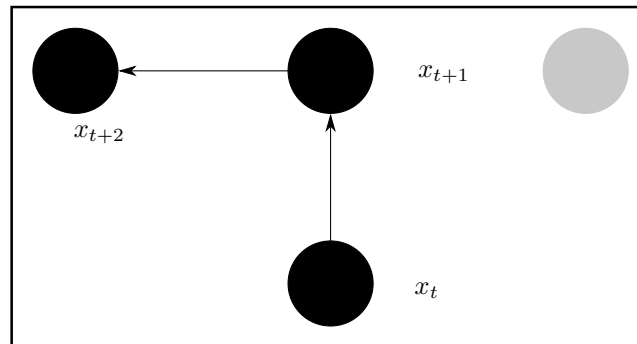


Abbildung 4.3.: Prinzip der Ermittlung von Navigationsanweisungen

gilt. In diesem Fall hat der Nutzer den zuvor errechneten Pfad verlassen. Um ihn aufgrund seiner neuen Position zum Ziel zu führen, wird eine neue Route errechnet, wobei der Startpunkt für die Berechnung die aktuelle Position x_t ist.

Auslösen von Navigationsanweisungen

Im Navigationsszenario des Gebäudes wurden drei unterschiedliche Situationen ermittelt, in denen eine Navigationsanweisung notwendig ist.

Zum einen, falls der Nutzer um eine Ecke gehen soll, zum anderen, wenn er Treppen oder Fahrstühle nutzen muss, um die Etage zu wechseln.

Anhand der aktuellen Position und der zwei nächstfolgenden Knoten im Pfad wird entschieden, welche Anweisung genutzt werden muss.

Für den Fall des Richtungswechsels wird geprüft, ob die Strecken $\overline{x_t x_{t+1}}$ und $\overline{x_{t+1} x_{t+2}}$ in einem gewissen Winkel zueinander stehen. Die Richtung wird aus dieser Information extrahiert, anschließend erfolgt die Anweisung, dass der Nutzer die Richtung wechseln muss (siehe Abbildung 4.3).

Im Fall der Treppe beziehungsweise des Fahrstuhls wird überprüft, ob die folgenden Knoten im Pfad zu einem dieser beiden Typen gehören. In dem Fall wird eine Meldung ausgegeben in welche Etage der Nutzer sich fortbewegen soll.

Trifft keiner dieser Fälle zu, wird keine Meldung ausgegeben.

4.3. Ausgabe von Navigationsanweisungen

Eine einfache Form der Darstellung des Pfades ist diesen auf einer Übersichtskarte des Gebäudes darzustellen. Die Route durch das Gebäude besteht aus einer Folge von Punkten. Zwischen ihnen kann eine Linie gezeichnet werden, um zu verdeutlichen, dass dieser Weg gegangen werden soll. Führt die Route über mehrere Etagen soll immer nur die Etage angezeigt werden, in der sich der

Benutzer zur Zeit befindet. Es kann weiter unterschieden werden, ob die Übersichtskarte statisch ist oder ob sie sich, ähnlich wie bei Geräten in Fahrzeugen, mit dem Nutzer mitdreht [But+01].

Eine weitere Möglichkeit ist es die Anweisungen durch eine Text to Speech (TTS) Software zu lösen. Dabei werden Befehle, die in Textform vorliegen, in Sprache übersetzt und über Lautsprecher am mobilen Gerät wiedergegeben [Au10, S. 61 ff.]. Da in dieser Arbeit eine Referenzimplementierung auf Android erfolgt, kann die seit Version 1.6 vorhandene TTS Software direkt genutzt werden.

5. Prototypische Umsetzung

In diesem Kapitel wird die prototypische Umsetzung der Lokalisierung und Navigation auf einem mobilen Endgerät beschrieben. Zunächst wird auf die verwendete Plattform eingegangen und anschließend notwendige Implementierungsdetails erläutert.

Die in Abschnitt 3 und 4 beschriebenen Ansätze lassen sich auf ein ganzes Gebäude anwenden. Die Implementierung erfolgte aus Zeitgründen jedoch nur im zweidimensionalen Raum und wird somit nur für die Ortung auf einer Etage beschrieben.

5.1. Entwicklungsumgebung

Zur Ortung sollen Magnetfeld- und Lichtsensor eingesetzt werden. Weiterhin muss die jeweils errechnete Route dem Nutzer visualisiert werden. Aufgrund des größeren Displays wurde sich dafür entschieden die Implementierung auf einem Tabletcomputer vorzunehmen.

In den folgenden Abschnitten wird kurz dargelegt für welche Entwicklungsumgebung sich für die prototypische Umsetzung entschieden wurde.

5.1.1. Softwareplattform

Die Implementierung der Software erfolgt auf einem Android basierendem Tabletcomputer. Das Software Development Kit (SDK) von Android erlaubt es verschiedenste Sensoren des Gerätes auszulesen. In dieser Arbeit wurde die Lokalisierung mit Hilfe von Magnetfeldern und Lichtfingerprints realisiert. Das iOS System von Apple hingegen lässt es nur zu, die Magnetfeldwerte des Sensors auszulesen. Der andere genutzte Sensor ist im Apple SDK nur über private Methoden des Application Programming Interface (API) zugänglich, welche es verhindern würden eine solche Anwendung im Apple AppStore zu verkaufen.⁵

Es existieren zur Zeit weitere Betriebssysteme für mobile Geräte. Der Fokus in dieser Arbeit liegt jedoch nicht auf der Auswahl einer optimalen Plattform, sondern auf der Überprüfung des vorgestellten Ansatzes zur Lokalisierung; darum wurden keine weiteren Geräte untersucht.

Die Software für Android wird in der Sprache Java geschrieben, wobei der produzierte Bytecode nicht kompatibel zur virtuellen Maschine von Oracle ist, das bedeutet, dass Android Anwendungen

⁵Nach „iPhone Developer Program License Agreement“, Abschnitt 3.3.1, Abruf 16. Februar 2012

zunächst nicht auf einer Standard-VM lauffähig sind. Zum Entwickeln und der Fehlersuche kann Eclipse mit einem Plugin für Android verwendet werden.

Zur Entwicklung der Ortungslösung wurden keine Bibliotheken, die nicht im Standard-SDK enthalten sind, genutzt.

5.1.2. Endgerät

Das verwendete Endgerät ist ein Motorola Xoom; zum Zeitpunkt des Erstellens dieser Arbeit war darauf die Android Version 3.2 installiert. Das Gerät besitzt folgende, für die Umsetzung relevanten, Hardwareeigenschaften⁶:

- 1 GHz Dual Core Prozessor
- 1 GB DDR2 Arbeitsspeicher
- Umgebungslicht- und Magnetfeldsensor, Barometer

5.2. Softwaredesign

In dieser Arbeit gilt es eine Ortungs- und Navigationslösung zu entwickeln. Die Kapitel zu den jeweiligen Strategien bezogen sich auf den dreidimensionalen Raum. Die prototypische Umsetzung berücksichtigt allerdings nur den zweidimensionalen. Das bedeutet, dass die Erkennung von Etagenwechseln nicht untersucht wurde.

5.2.1. Anforderungen

Um die Funktionalität des Systems zu überprüfen, werden an die Software folgende Anforderungen gestellt:

- Erstellung einer Fingerprintkarte für Magnetfelder und Lichtintensitäten
- Implementierung eines PF unter Nutzung der Fingerprintkarte
- Ortung mittels Magnetfeld- und Lichtsensor
- Implementierung einer Navigation
- Visualisierung der Position und des Pfades auf einer Übersichtskarte

⁶Nach <http://mediacenter.motorola.com/Fact-Sheets/Motorola-XOOM-Fact-Sheet-3537.aspx>, Abruf 16. Februar 2012

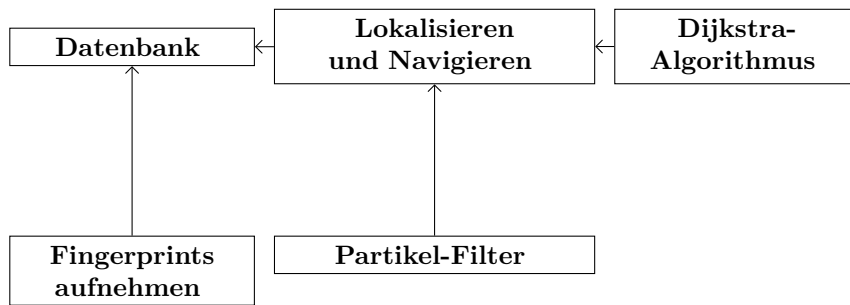


Abbildung 5.1.: Komponenten der prototypischen Umsetzung

5.2.2. Systemstruktur

Der entwickelte Prototyp umfasst unter anderem die Möglichkeit, auf dem mobilen Gerät Fingerprints für den Magnetfeld- und Lichtsensor zu erstellen. Weiterhin existiert eine Implementierung des PF um die zuvor beschriebene Ortung zu testen. Verknüpft mit der Ortung ist das Navigationsmodul, welches es dem Nutzer ermöglicht eine Route von der aktuellen Position zu einer gewünschten Stelle im Gebäude berechnen und visualisieren zu lassen.

In Abbildung 5.1 sind die Komponenten der prototypischen Umsetzung visualisiert. Sowohl die Komponente zur Erstellung der Fingerprints, als auch die Lokalisierungs Komponente nutzen eine gemeinsame Fingerprintdatenbank.

Das Datenbanklayout ist in Abbildung 5.2 aufgeführt. Diese Struktur wurde in dem Android Projekt im JavaScript Object Notation (JSON)-Format realisiert. JSON Daten sind Textdaten, somit eignet sich diese Variante um die benötigten Daten zum Beispiel über ein Netzwerk zu beziehen. Es wäre dann nicht mehr notwendig die Fingerprints direkt auf dem Gerät aufzunehmen, auf dem sie entstanden sind. Bei der Nutzung von Messdaten, die nicht auf dem selben Gerät aufgenommen wurden, gälte es zu überprüfen, ob die Messwerte im selben Spektrum liegen.

Ein Gebäude besteht aus mindestens einer Etage. Jede Etage wird durch eine Menge von Wänden beschrieben, welche als Linien dargestellt sind. Weiterhin sind in einer Etage beliebig viele Fingerprints vorhanden. Ein Fingerprint besteht aus der Position im Zustandsraum und den hinterlegten Daten. Pro Fingerprintposition können maximal vier Datensätze mit den Messdaten hinterlegt sein. In diesem Prototyp werden bei der Ortung verschiedene Ausrichtungen des Tablets beachtet. Dabei werden insgesamt vier Himmelsrichtungen unterschieden, wofür jeweils die Daten hinterlegt sind. Das Layout entspricht der Definition eines Fingerprints aus Formel 3.8.

5.3. Aufnehmen der Fingerprints

In der prototypischen Umsetzung werden die Messdaten für die Fingerprints direkt auf dem Gerät aufgenommen und nur dort gespeichert. Dieses Modul dient dazu die Daten einfach für die jeweilige Position aufzunehmen und in einer lokalen Datenbank zu speichern.

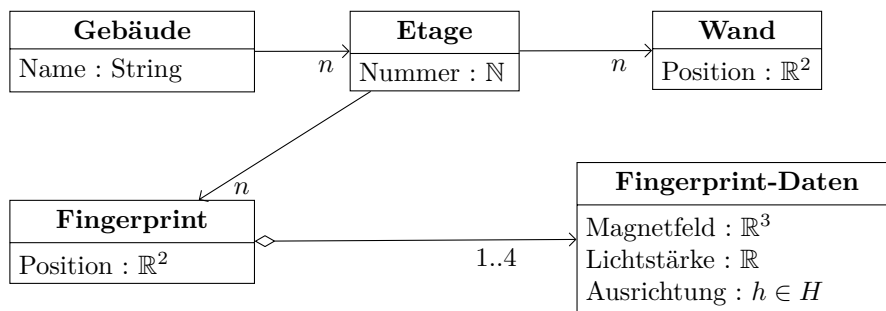


Abbildung 5.2.: Datenbankschema für die Speicherung von Fingerprintdatensätzen

Grafischer Hauptbestandteil ist eine Karte des Gebäudegrundrisses. Auf dieser werden die zuvor konfigurierten Positionen der Fingerprints dem Nutzer als Kreise angezeigt. In dieser Umsetzung waren die Positionen zuvor in die eingangs erwähnte Datenbank geschrieben worden; an dieser Stelle werden demnach nur noch die Messdaten hinterlegt.

Das Android Framework bietet dem Programm die Möglichkeit über neue Sensorereignisse informiert zu werden. Während dem Benutzer die Übersichtskarte angezeigt wird, werden im Hintergrund die Messdaten für Magnetfelder und Lichtintensität übergeben. Diese sind im Java Datentyp *double* gespeichert. Die Einheiten sind für jene Magnetfelder μT und Lux für die Lichtstärke. Beim Auswählen eines Fingerprints werden die aggregierten Messdaten gespeichert.

Wie in Abschnitt 3.2.6 erläutert, streut der Messwert vor allem des Magnetfeldsensors um den tatsächlichen Wert. Um diese Schwankungen auszugleichen, wird in der Fingerprint Datenbank nur ein Mittelwert der Sensordaten berechnet, welcher durch die Formel

$$\vec{m}_g = \frac{\sum_{i=1}^N \vec{m}_i}{N} \quad (5.1)$$

beschrieben ist. Dabei ist N die Anzahl der Messwerte, welche für das Mittel berücksichtigt werden sollen. In der Umsetzung wurden die sieben letzten Messwerte berücksichtigt; diese Anzahl reduziert die Streuung der Werte, sorgt aber beim Positionswechsel dafür, dass die alten Messwerte nicht zu lange den Mittelwert beeinflussen. Der Nutzer kann in diesem Teilprogramm für alle Positionen der Fingerprints die Messdaten für die vier Himmelsrichtungen hinterlegen.

Die Positionierung der Fingerprints erfolgte in der Umsetzung anhand der Pixelposition in der Übersichtskarte. Um die reelle Position in Metern zu erhalten, muss der Umrechnungsfaktor von Pixelpositionen in tatsächliche Koordinaten bestimmt werden. Dazu ist es notwendig, dass der Übersichtsplan maßstabsgetreu ist. Für diesen Faktor kann zum Beispiel eine Wand mit einem Zollstock ausgemessen und diese Länge in Relation zu der Länge im Übersichtsbild gesetzt werden. Dabei entsteht ein Verhältnis, das die Länge eines Pixels in Metern angibt. Bei der Umrechnung der Koordinaten wird dieses anschließend verwendet.

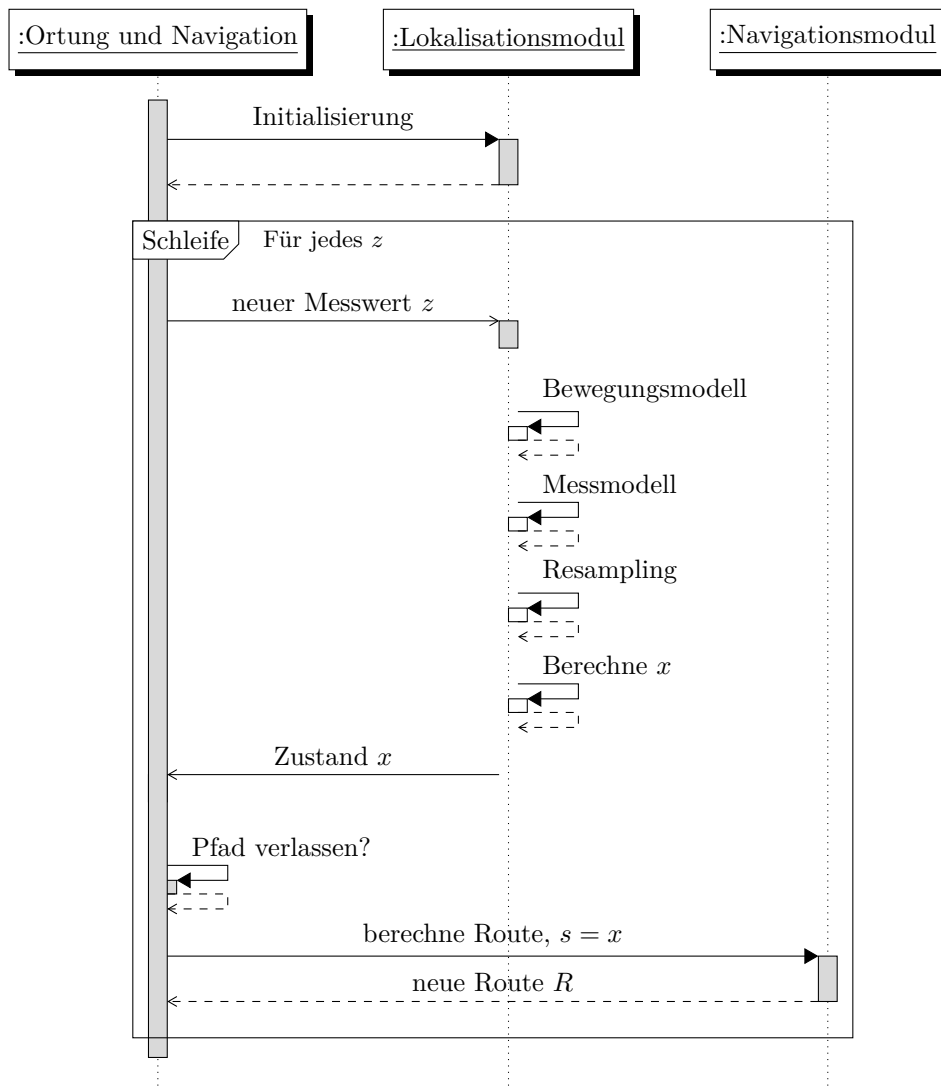


Abbildung 5.3.: Sequenzdiagramm der Lokalisierung

5.4. Lokalisationsmodul

Das Lokalisierungsmodul entspricht der Implementierung des in Abschnitt 3.2 vorgestellten PF. Die Umsetzung und noch nicht erläuterte Punkte wie die Filterinitialisierung, werden folgend kurz beschrieben.

In Abbildung 5.3 ist das Sequenzdiagramm für das Lokalisationsmodul aufgeführt. Zunächst muss der Filter initialisiert werden (siehe Abschnitt 5.4.1). Anschließend wird für jeden eingetroffenen Messwert der Algorithmus durchlaufen. Am Ende hat der Filter einen neuen Zustand geschätzt. Falls der Nutzer während der Navigation den zuvor berechneten Pfad verlassen hat, kann ein neuer auf Basis der neuen Position berechnet werden.

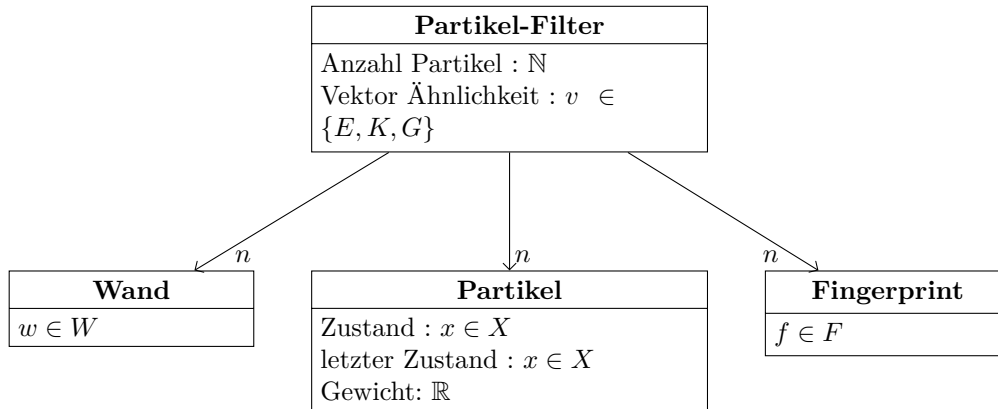


Abbildung 5.4.: Klassendiagramm des Partikel-Filters

Die Abbildung 5.4 zeigt das Klassendiagramm der Umsetzung des PF. Die Wände und die Fingerprints wurden aus der Datenbank entnommen (siehe Abbildung 5.2). In der Evaluation soll der Einfluss der Anzahl der Partikel und des verwendeten Ähnlichkeitsmaßes ermittelt werden; folglich ist der PF damit parametrisiert. Ein Partikel ist entsprechend der Definition aus Formel 2.16 aufgebaut. Er wurde noch um den zuletzt eingenommenen Zustand erweitert. Dieser wird im Messmodell bei der Überprüfung, ob der Partikel eine Wand beim Bewegungsmodell durchquert hat, benötigt.

5.4.1. Partikel-Filter Initialisierung

Vor dem Lokalisieren muss der PF initialisiert werden. Da zu diesem Zeitpunkt noch keine Positionsinformationen zur Verfügung stehen, müssen die Partikel über den gesamten Zustandsraum verteilt werden. Dies geschieht in einer Gleichverteilung und kann durch den Algorithmus 5.1 erfolgen.

Algorithmus 5.1 Partikel-Filter Initialisierung (w_m, h_m, N)

```

1:  $P \leftarrow \emptyset$ 
2:  $n_h \leftarrow \sqrt{N \cdot \frac{h_m}{w_m}}$  ▷ Anzahl der Zeilen
3:  $n_w \leftarrow \frac{N}{n_h}$  ▷ Anzahl der Spalten
4:  $s \leftarrow \frac{w_m}{n_w}$  ▷ Abstand zwischen den Partikeln
5:  $c \leftarrow \lceil n_w + 0,5 \rceil$  ▷ Abrunden der Spalten
6:  $r \leftarrow \lceil n_h + 0,5 \rceil$ 
7: for  $i = 0$  to  $c - 1$  do
8:   for  $j = 0$  to  $r - 1$  do
9:      $P \cup \langle i \cdot s, j \cdot s, \frac{1}{c \cdot r} \rangle$  ▷  $\langle p_x, p_y, w \rangle$ 
10:   end for
11: end for
    
```

Dieser Algorithmus berechnet aus der Breite und Höhe der Übersichtskarte und der benötigten Anzahl an Partikeln die Startkonfiguration dieser. Dabei wird jedem Partikel die gleiche Startge-

wichtung zugewiesen. Die Liste P enthält die Partikel mit ihrer zweidimensionalen Position und dem Startgewicht.

5.4.2. Ausführen eines Filterschritts

Bei jedem Eintreffen neuer Messdaten wird ein Partikel-Filter-Schritt durchlaufen, dieser ist in Abschnitt 2.5.3 beschrieben. Dieser wurde, bis auf die Umsetzung im dreidimensionalen Raum, auf die dort beschriebene Weise implementiert.

Um die Sensordaten auf einem Android System auszulesen, muss sich die Anwendung mittels *Listener-Objekt* am *SensorManager* anmelden. Dabei kann angegeben werden, wie oft neue Sensordaten der Anwendung zur Verfügung gestellt werden sollen.⁷ Die genaue Aktualisierungsrate kann daher nicht bestimmt werden. In der erstellten Implementierung wird jedoch alle 500 ms eine neue Iteration begonnen. In dieser Zeit werden die Sensordaten gesammelt und anschließend ein Mittelwert daraus gebildet.

Die einzelnen Schritte des Filters werden im Folgenden näher beschrieben.

Konfigurationsmöglichkeiten des Algorithmus

Ein PF ist zunächst ein Filterkonzept, wobei einige Teile des Algorithmus austauschbar sind. Um das Verhalten und die Genauigkeit des Algorithmus zu untersuchen, wurde dieser für bestimmte Parameter konfigurierbar gestaltet.

Dazu zählt die Anzahl der verwendeten Partikel, wobei dort eine beliebige natürliche Zahl genutzt werden kann. Bei dem Vergleich von Vektoren wurde ein probabilistischer Ansatz, die Kosinus-Ähnlichkeit und die euklidische Distanz implementiert (siehe Abschnitt 3.2.6).

Weiterhin ist es möglich, verschiedene Sensoren in die Ortung einzubeziehen. Dazu zählt der Magnetfeldsensor, der Lichtsensor und die Kombination beider. Die Beachtung von Wänden im Messmodell sowie das Berücksichtigen der Himmelsrichtung wurden ebenso konfigurierbar gestaltet (siehe Abschnitt 3.2.6).

Bewegungsmodell

Der PF basiert sowohl im Mess- als auch im Bewegungsmodell auf gaußverteilten Zufallsvariablen. Auf dem Android System steht ausschließlich das Standard SDK zur Verfügung. Dieses bietet jedoch nur die Möglichkeit Zufallszahlen mit $\mu = 0$ und $\sigma^2 = 1$ zu erzeugen.

In Algorithmus 5.2 ist die Polar-Methode zum Erzeugen von gaußverteilten Zufallsvariablen aufgeführt [Knu97, S. 122 f.].

Die Ausgabe sind zwei gaußverteilte Zufallsvariablen nach $\mathcal{N}(\mu, \sigma^2)$.

⁷vgl. <http://developer.android.com/reference/android/hardware/SensorManager.html>, Abruf 16. Februar 2012

Algorithmus 5.2 Polar-Methode zur Erzeugung von gaußverteilten Zufallsvariablen (μ, σ)

```

1: repeat
2:    $u_1 \leftarrow [0, 1]$  ▷ Gleichverteilte Zufallszahl
3:    $u_2 \leftarrow [0, 1]$ 
4:    $v_1 \leftarrow 2 \cdot u_1 - 1$  ▷ Zufallszahl zwischen  $-1$  und  $1$ 
5:    $v_2 \leftarrow 2 \cdot u_2 - 1$ 
6:    $s \leftarrow v_1^2 + v_2^2$ 
7: until  $s < 1 \wedge s \neq 0$ 
8:  $x_1 \leftarrow \mu + \sigma v_1 \sqrt{\frac{-2 \ln s}{s}}$ 
9:  $x_2 \leftarrow \mu + \sigma v_2 \sqrt{\frac{-2 \ln s}{s}}$ 

```

In diesem Modell werden die Partikel innerhalb des Zustandsraum bewegt. Zunächst werden die Bewegungsparameter für jeden Partikel einzeln bestimmt. Dazu gehören die neue Geschwindigkeit (siehe Formel 3.3) und die Richtung (siehe Formel 3.5). In der Implementierung wurde eine Ortung im Zweidimensionalen umgesetzt, deshalb wurde zum Verschieben der Partikel die Formel 2.22 ohne die dritte Dimension benutzt. Das dort verwendete Δt entspricht der Dauer zwischen zwei Messwerten. Das Rauschen der Messwerte wird durch $\mathcal{N}(0, 10^{-1})$ modelliert. Nach diesem Schritt wurde jeder Partikel aufgrund des Bewegungsmodells in einen neuen Zustand versetzt, welcher im Messmodell bewertet wird.

Messmodell

Wie zuvor genannt, ist der Algorithmus deshalb konfigurierbar, um die einzelnen Teile und ihre unterschiedlichen Funktionsweisen miteinander vergleichen zu können. In Algorithmus 5.3 ist das Messmodell beschrieben.

Algorithmus 5.3 Umsetzung des Messmodells (p, F, z)

```

1:  $x \leftarrow$  Zustand  $(p, F)$  ▷ Zustand, in dem sich ein Partikel befindet
2:  $g \leftarrow 1$ 
3: for all  $t \in z$  do ▷ Für jeden Messtyp den  $z$  enthält
4:   if  $t =$  Magnetfeld then
5:      $g \leftarrow g \cdot P(m | x)$  ▷ Wahrscheinlichkeit für Magnetfeld
6:   end if
7:   if  $t =$  Lichtintensität then
8:      $g \leftarrow g \cdot P(l | x)$  ▷ Wahrscheinlichkeit für Lichtintensität
9:   end if
10: end for
11: if  $\text{NutzeWandAspekt} \wedge \text{WandDurchquert}(p)$  then
12:    $g \leftarrow 0$ 
13: end if
14:  $w_p \leftarrow g$  ▷ Neues Gewicht für Partikel

```

In dieser Umsetzung ist bereits berücksichtigt, dass nur Magnetfelder, Lichtintensitäten oder beide Werte genutzt werden. Die Implementierung von zum Beispiel $P(m | x)$ hängt vom gewählten

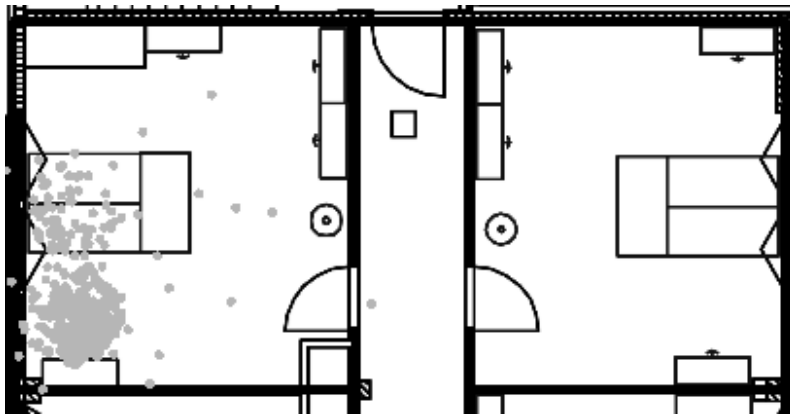


Abbildung 5.5.: Partikelverteilung nach dem Resampling

Vektorähnlichkeitsmaß ab. Die Wahrscheinlichkeiten dafür sind jeweils in Formel 3.11 und 3.12 erläutert.

Resampling Verfahren

Das in dieser Arbeit umgesetzte Resampling Verfahren entspricht dem in Abschnitt 2.5.6 vorgestellten Resampling-Algorithmus.

Die Abbildung 5.5 zeigt eine Verteilung der Partikel nach dem Resampling. Dabei ist zu erkennen, dass sich die Partikel auf zwei Fingerprints verteilt haben, weil laut Messmodell die Wahrscheinlichkeit dieser Zustände am höchsten ist.

5.4.3. Ermittlung des Systemzustands

Um den Systemzustand nach dem Filterschritt auszugeben, wurde die Formel 2.19 genutzt. Diese gibt die Wahrscheinlichkeit an, mit welcher sich ein System im Zustand x_t befindet.

x_t ist in diesem Fall eine Fingerprint Position. Um dies zu berechnen, musste für jeden Fingerprint bestimmt werden, welche Partikel sich in diesem befinden. Dafür wurde für jeden Partikel der Fingerprint gesucht, welcher ihm am nächsten ist. Sowohl der Partikel als auch der Fingerprint besitzen eine Position im Raum. Die Distanz zwischen beiden ergibt sich aus der euklidischen Distanz.

Gleichzeitig wird eine Liste des Gesamtgewichts für jeden Fingerprint geführt. Das Gewicht eines Partikels wird anschließend auf das Gesamtgewicht des jeweiligen Fingerprints addiert. Der Fingerprint mit dem höchsten Gesamtgewicht stellt den vom PF geschätzten Systemzustand dar. Dieses Vorgehen ist in Algorithmus 5.4 beschrieben. Die Position des Systems kann anschließend in der Navigation verwendet werden.

Algorithmus 5.4 Berechnung des Systemzustands (P, F)

```
1: for all  $f \in F$  do ▷ Initialisierung der Gewichtsliste
2:    $\pi[f] \leftarrow 0$ 
3: end for
4: for all  $p \in P$  do ▷ jeder Partikel
5:    $f \leftarrow \text{Zustand}(p, F)$ 
6:    $\pi[f] \leftarrow \pi[f] + w_p$  ▷ Gewicht des Partikels wird aufaddiert
7: end for
8:  $x \leftarrow \text{Max}(\pi)$ 
```

5.4.4. Zuordnung zwischen Partikelposition und der Visualisierung

Neben der aktuell geschätzten Position des Gerätes wurde zum besseren Verständnis des Partikel-Filters gleichfalls die einzelnen Partikel visualisiert. Dies ermöglichte es, die Ausbreitung im Raum und die Verteilung an sehr wahrscheinlichen Fingerprints zu studieren.

Die Position der Partikel bezog sich, genau wie die der Fingerprints, auf ihre Pixelposition in der Übersichtskarte. Diese Karte hat in der Regel eine andere Auflösung als das genutzte Gerät. Zu diesem Zweck müssen die Positionen der Partikel, an der korrekten Stelle auf dem Gerätedisplay dargestellt werden.

Meist ist das Seitenverhältnis zwischen Bildschirm und Karte unterschiedlich, dadurch muss zur korrekten Darstellung sowohl ein Skalierungsfaktor als auch ein Offset errechnet werden. Dieses Offset wird benötigt, wenn beispielsweise durch die Skalierung der Karte schwarze Ränder hinzugefügt werden.

In Algorithmus 5.5 ist die Berechnung dieser drei Werte aufgeführt. w_d, h_d ist die Größe des Bildschirms und w_r, h_r , die der Übersichtskarte.

Algorithmus 5.5 Berechnung der Skalierungsfaktoren bei der Darstellung der Partikel (w_d, h_d, w_r, h_r)

```
1:  $s_w \leftarrow \frac{w_d}{w_r}$ 
2:  $s_h \leftarrow \frac{h_d}{h_r}$ 
3:  $s \leftarrow \min(s_w, s_h)$ 
4:  $o_w \leftarrow \frac{1}{2} \cdot (w_d - (w_r \cdot s))$ 
5:  $o_h \leftarrow \frac{1}{2} \cdot (h_d - (h_r \cdot s))$ 
```

Die Ausgabe ist der Skalierungsfaktor s , der Offset für die y-Achse o_h und der x-Achse o_w .

5.5. Navigationsmodul

Der in diesem Prototyp implementierte Dijkstra-Algorithmus konnte, bis auf die PriorityQueue, aus dem Algorithmus 2.3 abgeleitet werden. Voraussetzung für den Algorithmus ist ein gewichteter Graph. In Abschnitt 4.1.1 wird ein Algorithmus erklärt, um aus den Fingerprints einen Graphen zu berechnen. Falls beim Ermitteln des Nachbarn keine Resultate gefunden werden, bedeutet dies,

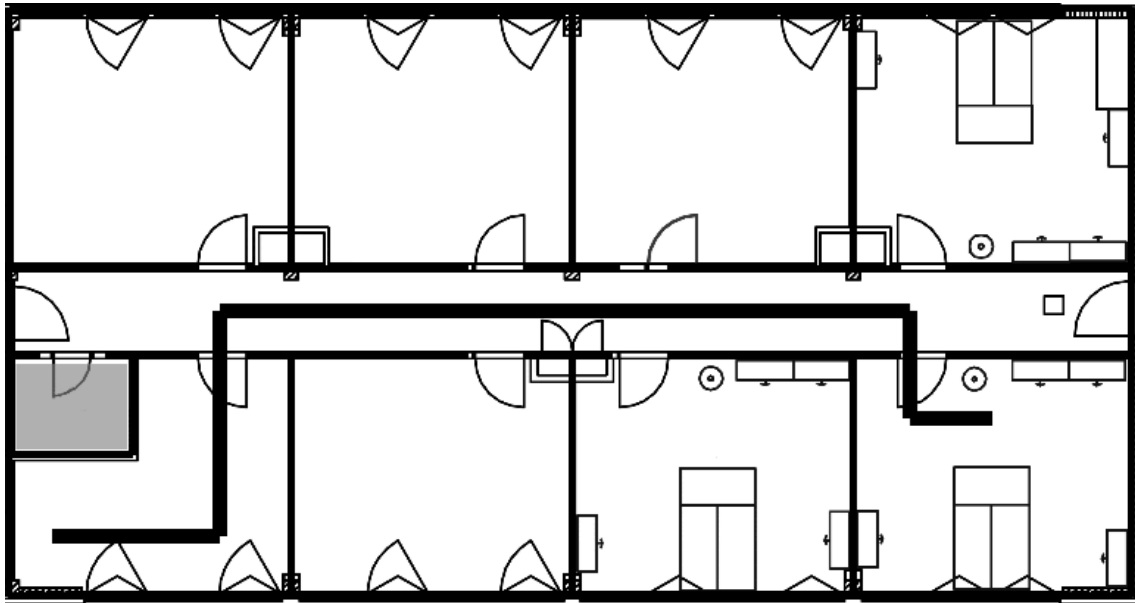


Abbildung 5.6.: Bildschirmfoto einer Route in der dritten Etage im Testgebäude

dass der Graph nicht vollständig verbunden ist. In diesem Fall wird in der Implementierung die Grapherstellung abgebrochen. Die Positionen der Fingerprints und der Wände sind demnach falsch konfiguriert; eine spätere Navigation kann nicht mehr erfolgen.

Wurden die Karteninformationen korrekt gepflegt, kann der Graph erstellt werden. Nach der Berechnung des kürzesten Pfades steht dieser in einer geordneten Liste zur Verfügung. Jeder Eintrag ist demnach ein Knoten und somit ebenso eine Fingerprintposition im Graph. In dieser Umsetzung wurde der Pfad als Aneinanderreihung der Liniensegmente auf der Übersichtskarte realisiert. In der Abbildung 5.6 ist eine beispielhafte Route in dem Testgebäude dargestellt.

6. Evaluation der Ortungs- und Navigationslösung

In diesem Kapitel werden die Ergebnisse des PF und der Navigation vorgestellt. Zunächst wird erläutert, wie die Messdaten aufgenommen wurden. Anschließend wird kurz erläutert, wie die große Anzahl an Konfigurationen des Algorithmus miteinander verglichen werden kann. Diese Konfigurationen werden im Hinblick auf ihren Einfluss auf die Präzision untersucht.

In der Fehlerbetrachtung werden mögliche negative Einflüsse auf das Ergebnis der Ortung beschrieben. Die gewonnenen Erkenntnisse werden mit den Ergebnissen anderer Lokalisierungstechniken verglichen. Der Vergleich erfolgt dabei über die erreichte Genauigkeit und ihren Installationsaufwand. Weiterhin wird die Funktionalität des Navigationsalgorithmus kurz demonstriert.

6.1. Methodik

Die Messungen wurden wieder in dem Bürokomplex des Technologie- und Gründerzentrums Spreeknäe in Berlin durchgeführt. Um die Ortungsgenauigkeit zu bestimmen, wurden insgesamt fünf Testläufe in der dritten Etage absolviert. Dabei wurden vor jedem die Fingerprints aufgenommen. Wie in Abschnitt 3.1.2 beschrieben, waren die Ergebnisse für die Büroräume aus der dritten Etage die besten, weshalb der Versuch dort vollzogen wurde.

Im Anschluss wurde der PF zur Lokalisierung gestartet. Dafür wurden beliebige Pfade durch die dritte Etage gegangen, wobei an einem Punkt verweilt wurde, damit der Algorithmus zu einem Zustand konvergieren konnte. Anschließend wurde in einer Datei vermerkt, an welcher Position das mobile Gerät sich gerade befindet. Daraufhin wurde zu einem neuen Punkt gegangen.

Bei zwei der fünf Messungen wurde die gesamte Etage zum Orten genutzt. Diese ist zirka 300 m^2 groß und es wurden 52 Fingerprints aufgenommen. Für die drei weiteren Messungen wurde die Hälfte der Etage genutzt, welches insgesamt 27 Fingerprints entspricht.

6.1.1. Aufzeichnen der Messdaten

Der PF umfasst in der vorgestellten Variante mehrere Konfigurationsmöglichkeiten. Diese sind die Beachtung von Wänden, das Hinzuziehen der Himmelsrichtung für eine bessere Auswahl der

Fingerprints, die Nutzung von verschiedenen Sensoren, Verwendung verschiedener Vektorähnlichkeitsmaße und die Anzahl der Partikel zur Approximation der Position: insgesamt sind dies 144 Kombinationen. Wie zuvor genannt, wurden fünf Sätze von Messwerten aufgenommen. Zum einen ist die Anzahl der Kombinationen zu hoch, um für jede eine eigene Ortung im Testgebäude durchzuführen zum anderen ist dies gar nicht von Vorteil, da für jede Ortung andere Messwerte entstehen würden. Es wäre demnach nicht möglich die einzelnen Messungen bei unterschiedlicher Konfiguration miteinander zu vergleichen.

Damit nicht für jede Konfiguration eine neue Ortung durchgeführt werden muss, sollte die Schätzung der jeweiligen Position ohne den Tabletcomputer simuliert werden können.

Zur Realisierung wurden während einer Messung die Sensordaten von Magnetfeld- und Lichtsensor, die Ausrichtung des Gerätes und ein Zeitstempel, wann diese Messung auftrat, in einer Datei gespeichert. Weiterhin wurden während der Ortung, wie in Abschnitt 6.1 beschrieben, die tatsächliche Position und ihr zeitliches Auftreten, gespeichert.

Es wurde eine weitere Software entwickelt, welche die gleiche PF-Implementierung wie auf dem mobilen Gerät nutzt. Messwerte und reale Positionen werden nach ihren Zeitstempeln sortiert und dem Algorithmus übergeben. Bei einer Positionsschätzung wird die echte Position des Gerätes (beziehungsweise die Position aus der Datei) zusammen mit der Schätzung des Algorithmus in eine Ausgabedatei zur späteren Auswertung geschrieben.

Die verschiedenen Konfigurationen wurden alle auf die gleiche Weise simuliert und die Ergebnisse gespeichert.

Pro Konfiguration des Algorithmus wurde anschließend der durchschnittliche Fehler berechnet. Der Root Mean Square Error (RMSE) wird in der Literatur meist als Wert für den durchschnittlichen Fehler genutzt. Dieser ist definiert durch:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N \|p_i^r - p_i^e\|^2} \quad (6.1)$$

N ist dabei die Anzahl der aufgenommenen Positionen für eine Konfiguration, p_i^r die tatsächliche und p_i^e die vom Algorithmus geschätzte Position.

6.1.2. Box-Plots zum Vergleich der Konfigurationen

Bei den später genutzten Diagrammen zur Visualisierung der Fehlerverteilung handelt es sich um Box-Whisker-Plots. Diese fassen statistisch stabile Kennwerte in einer Grafik zusammen. Hauptbestandteil ist die Box, in der 50 % der Daten liegen. Die horizontale Linie ist dabei der Median. Die so genannten Antennen repräsentieren Werte außerhalb des 50 % Bereiches. Es können auch Werte außerhalb der Antenne sich befinden; dies sind statistische Ausreißer.

Wie bereits beschrieben, werden die verschiedenen Konfigurationen des Algorithmus auf ihre Auswirkung untersucht. Die Daten, welche visualisiert werden, sollen kurz am Beispiel der Abbildung 6.1 dargestellt werden. Diese Grafik ist nach der Beachtung von Wänden, später nur noch Wandaspekt genannt, innerhalb des Algorithmus aufgeteilt. In dem Box-Plot bei der Variable *Wand* sind alle Konfigurationen enthalten, bei dem diese Funktion genutzt wird. Darunter fallen die Kombinationen wie die Beachtung von Himmelsrichtungen und der genutzte Vektorvergleich. Der andere Plot zeigt die Konfigurationen, bei denen dieser Wandaspekt nicht genutzt wurde.

Pro Konfiguration entstehen eine Reihe von Messfehlern. Diese treten auf, wenn die tatsächliche Position mit der geschätzten verglichen wird. Für jede wird der RMSE nach Formel 6.1 gebildet. Die einzelnen RMSE Werte stellen die Fehlerverteilung in einem Box-Plot dar.

Diese Darstellung dient dem schnellen Ablesen von Einflüssen bestimmter Parameter, da statistische Kennwerte kompakt nebeneinander dargestellt werden.

6.2. Genauigkeit bei der Ortung

Zentraler Bestandteil einer Ortung ist ihre Genauigkeit und die Reaktivität bei Ortswechseln. In den folgenden Abschnitten werden die Konfigurationen hinsichtlich ihres Einflusses auf die Genauigkeit beschrieben und verglichen.

6.2.1. Wände zur Verbesserung der Ortung

In Abschnitt 3.2.4 wurde erläutert, dass eine Umgebungskarte genutzt werden kann, um ungültige Partikelbewegungen zu eliminieren. In Abbildung 6.1 sind die RMSE aller Simulationen, getrennt nach dem Wandaspekt, aufgeführt.

Der Median des RMSE liegt für den Wandaspekt bei 6,53 m und ohne diesen bei 6,41 m, welche sich kaum unterscheiden. Während der Aufzeichnung der Messdaten konnte jedoch festgestellt werden, dass bei der Nutzung des Wandaspektes der Filter Schwierigkeiten hatte Fehlschätzungen in darauffolgenden Schritten rückgängig zu machen.

Kommt es während der Ortung zu Fehlern, zum Beispiel wenn das Gerät durch den Algorithmus fälschlicherweise in einen anderen Raum vorhergesagt wird, ist es für den PF schwierig, diese Fehleinschätzung aufgrund von Wänden zu korrigieren. Nach dem in Abschnitt 2.5.5 vorgestellten Messmodell werden Partikel bei der Bewegung durch Wände mit 0 bewertet, ihre Wahrscheinlichkeit in eine Folgegeneration aufgenommen zu werden ist somit gering. Wird das mobile Gerät in einem falschen Raum vermutet, müssten die Partikel durch eine Wand bewegt werden, um zu einem späteren Zeitpunkt die korrekte Position wieder zu erreichen.

Die statistisch stabilen Werte aus dem Box-Plot ergeben keinen Unterschied, ob im Filter Wände berücksichtigt werden. Es kommt allerdings ohne diese Funktion zu einem geringeren maximalen Fehler und während der Ortung wurden fehlerhafte Positionierungen schneller ausgeglichen.

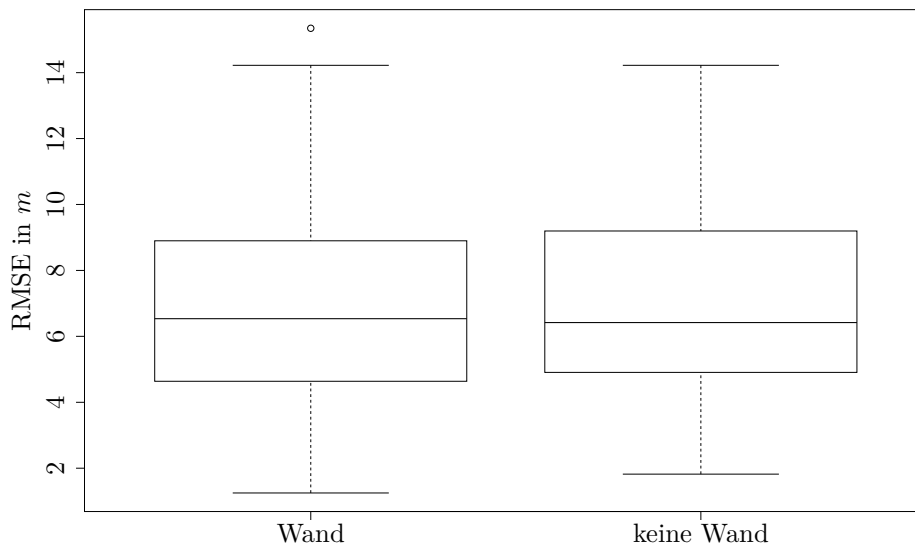


Abbildung 6.1.: Fehlerverteilung bei der Berücksichtigung von Wänden im Messmodell

6.2.2. Himmelsrichtung zum Ausgleich des Fehlers bei der Vektornormierung

In Abschnitt 3.2.6 wurde erläutert, dass für jeden Fingerprint ein Messwert für jede Himmelsrichtung aufgenommen wird, damit der vom Gerät falsch normierte Magnetfeldvektor ausgeglichen werden kann. In der Simulation lag beim Beachten der Himmelsrichtung der Median des Fehlers bei 6,43 m und ohne diese Funktion bei 6,54 m.

Die Abbildung 6.2 zeigt den Ortungsfehler unterschieden nach dieser Funktion. Der geringste RMSE lag bei beiden Konfigurationen auf einem sehr ähnlichen Niveau. Die minimalen Fehler betragen zum einen 1,38 m und zum anderen 1,24 m. Es kann in diesem Fall kein Einfluss dieser Funktion auf die Ortungsgenauigkeit festgestellt werden.

6.2.3. Vergleich von Vektorähnlichkeiten

Der Abschnitt 3.2.6 beschreibt ein Messmodell, wonach die Ähnlichkeit zwischen Messvektor und Fingerprint, auf dem sich ein Partikel befindet, errechnet wird. In dieser Arbeit wurden drei verschiedene Vektorähnlichkeiten implementiert.

Die Abbildung 6.3 visualisiert die Fehlerverteilung für die drei Vektorähnlichkeitsmaße. Der Median der Fehler beträgt bei der euklidischen Distanz 5,8 m, bei der gaußschen Ähnlichkeit 7,0 m und bei der Kosinus-Ähnlichkeit 6,8 m. Der Unterschied bei dieser Konfiguration zwischen dem besten und dem schlechtesten Ergebnis beträgt 1,2 m. Es kann davon ausgegangen werden, dass sich diese Differenz außerhalb eines Messfehlers befindet. Ebenfalls liefert der Wilcoxon-Signifikanztest, zur Überprüfung der Gleichheit zweier Stichproben, keine Übereinstimmung der Ergebnismengen.

Zunächst wurde davon ausgegangen, dass sich der probabilistische Ansatz besser eignet, da in diesem der Messfehler der Sensoren berücksichtigt werden kann. Es zeigte sich jedoch, dass mittels

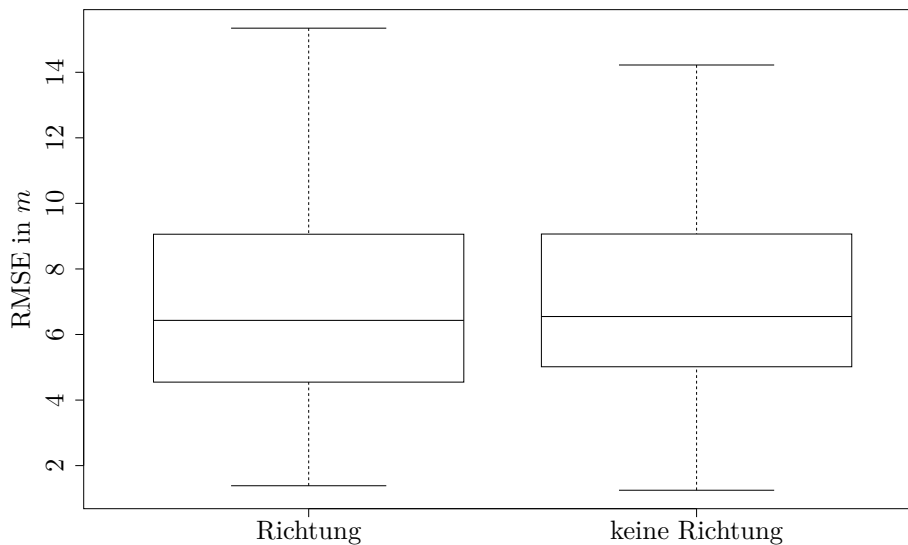


Abbildung 6.2.: Fehlerverteilung bei der Berücksichtigung der Himmelsrichtung

diesen Ansatzes es öfter zu wahrscheinlichsbasierten Überschneidungen von Fingerprints kam, was eine falsche Ortungsvorhersage zur Folge hatte.

Ein weiterer negativer Effekt tritt auf, wenn Messwerte zu keinem Fingerprint passen, weil zum Beispiel das Magnetfeld kurzfristig gestört ist oder sich die Lichtverhältnisse ändern. In diesem Fall ist die Übereinstimmung der Messwerte sehr gering, die Partikel werden niedrig bewertet und können im Resampling-Schritt mit einer hohen Wahrscheinlichkeit substituiert werden.

Bei der euklidischen Distanz und der Kosinus-Ähnlichkeit tritt dieser Effekt nicht auf, da selbst wenn sich Messwert und alle Fingerprints nicht ähneln, ihm rechnerisch einer am „ähnlichsten“ sein muss. Dieser erhält eine hohe Bewertung und überdauert den Resampling-Schritt.

Die Kosinus-Ähnlichkeit wurde als Vergleichsmaß genutzt, weil untersucht werden sollte, ob die Richtung eines Magnetfeldes wiedererkannt wird. Die erreichten Werte lassen jedoch den Schluss zu, dass die euklidische Distanz besser für dieses Szenario geeignet ist.

6.2.4. Nutzen von mehreren Sensoren

Der Einfluss der genutzten Sensoren auf die Ortungsgenauigkeit wird in Abbildung 6.4 visualisiert. Verglichen werden die Genauigkeit von Magnetfeld- oder Lichtfingerprints und die Kombination beider Sensoren. Der Median des Fehlers lag bei ausschließlich lichtbasierter Ortung bei 7,5 m, beim Magnetfeld bei 6,37 m und unter Nutzung beider Sensoren bei 5,93 m.

Trotz geringer Aussagekraft der Lichtintensität konnte eine, für diesen Sensor, akzeptable Genauigkeit erzielt werden. Diese Leistung hängt jedoch stark vom Einsatzort und von der Größe der zu ortenden Fläche ab. Die Nutzung des, im Vergleich, eindeutigeren Magnetfeldes konnte eine

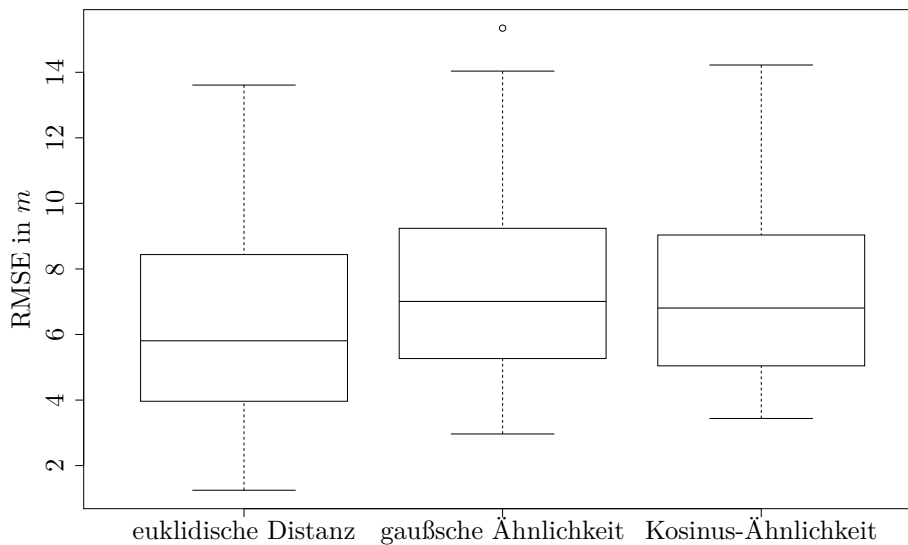


Abbildung 6.3.: Fehlerverteilung bei verschiedenen Vektorähnlichkeiten

Verbesserung der Ergebnisse liefern. Die Kombination beider Sensoren hat die Genauigkeit der Ortung im Schnitt nochmals verbessert.

Weil mobile Geräte oft beide Sensoren bereits enthalten, ist die Kombination beider Sensoren der Einzelnutzung vorzuziehen.

6.2.5. Einfluss der Partikelanzahl

Die Anzahl der genutzten Partikel zur Repräsentation der WDF hat einen Einfluss auf die Genauigkeit der Schätzung. Laut Literatur ist die Schätzung umso optimaler je mehr Partikel genutzt werden [Wid10, S. 29]. Diese kann aber nicht beliebig hoch sein, da durch eine steigende Anzahl an Partikeln der Rechenaufwand steigt. Gerade bei Echtzeitanwendungen ist es entscheidend einen Kompromiss zwischen beiden Faktoren zu finden. Weiterhin wird in der Literatur oft eine Menge von mindestens 500 Partikeln oder mehr vorgeschlagen (siehe [Wid10; KW08; HK09]).

Die Abbildung 6.5 zeigt den Einfluss der Anzahl der Partikel auf den mittleren Ortungsfehler. Die drei Konfigurationen ab 200 Partikeln besitzen einen ähnlichen Median. Die Partikelmenge bestehend aus 50 hat im Schnitt den höchsten Fehler.

Daraus ist abzuleiten, dass für den PF eine Mindestanzahl von Partikeln benötigt wird. Diese ist aber geringer als die meistens vorgeschlagene Menge. Der Grund für diesen Effekt könnte der kleine Zustandsraum sein, welcher von der Anzahl beziehungsweise der Auflösung der Fingerprints abhängt. Eventuell würde bei einem größeren Zustandsraum die Rolle der Partikelanzahl größer sein.

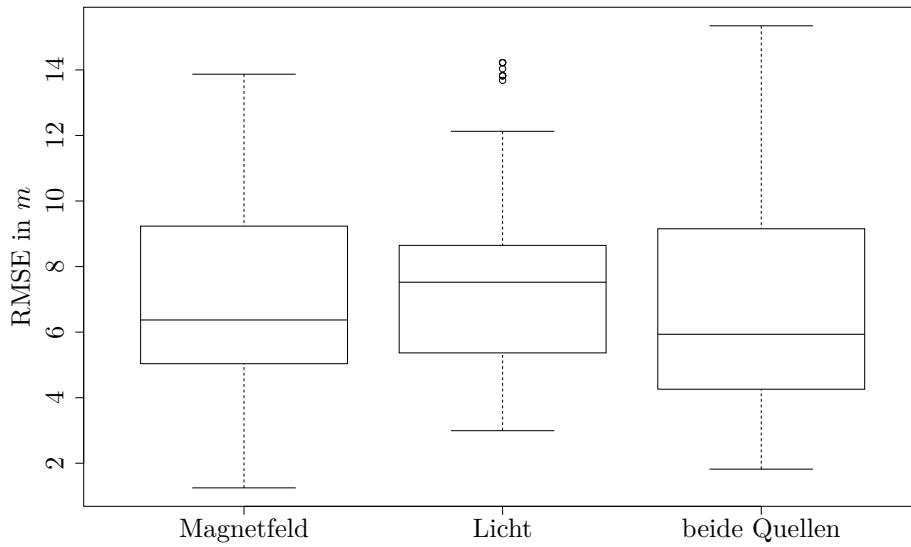


Abbildung 6.4.: Fehlerverteilung bei der Nutzung von verschiedenen Sensoren

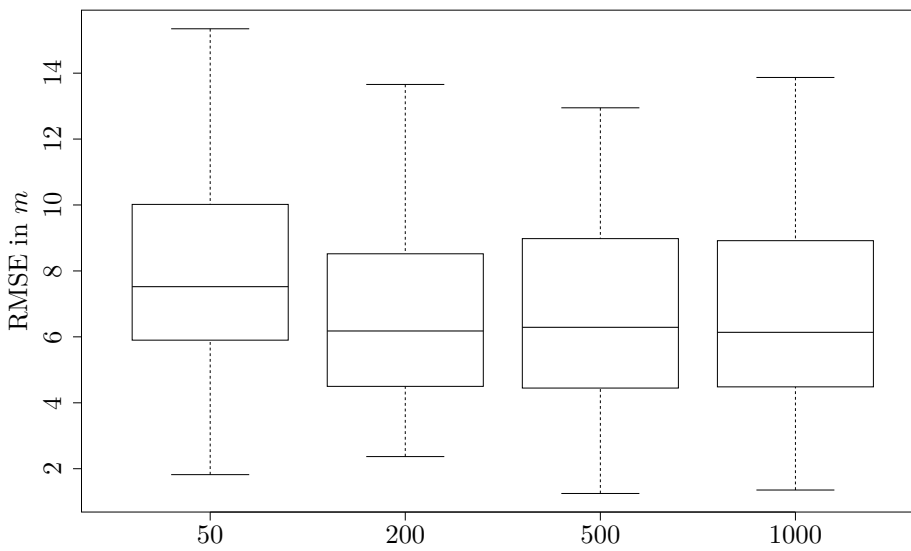


Abbildung 6.5.: Fehlerverteilung bei unterschiedlicher Anzahl an Partikeln

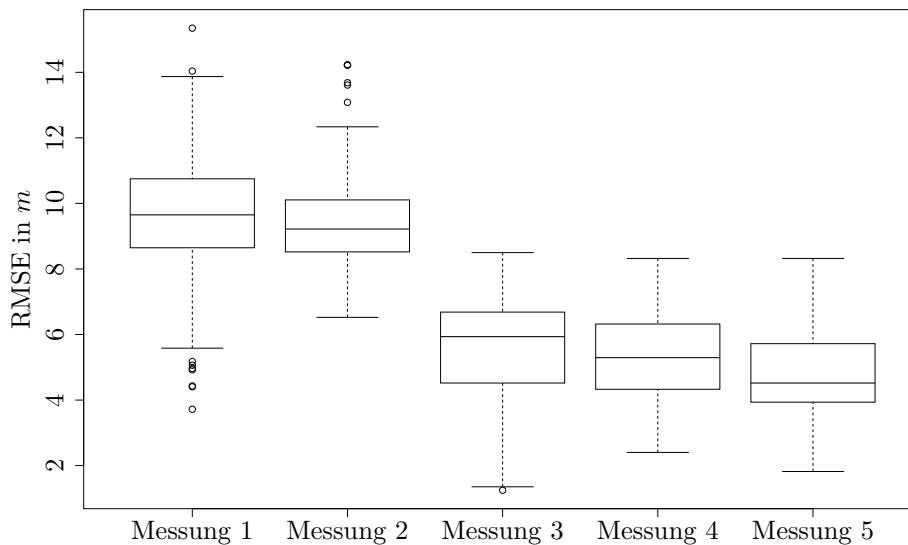


Abbildung 6.6.: Fehlerverteilung bei den verschiedenen Messungen

6.2.6. Fehler bei den unterschiedlichen Messungen

In den vorherigen Abschnitten wurden jeweils die aggregierten Messfehler aller fünf Messungen aufgeführt (siehe Abschnitt 6.1). In Abbildung 6.6 sind diese nach den einzelnen Messungen gelistet. Die Messungen 3 bis 5 haben im Schnitt die besten Ortungsergebnisse. Bei diesen Testdurchläufen wurde nur die halbe Etage für die Ortung verwendet. Mit steigender Anzahl an Fingerprints nimmt ihre Aussagekraft ab und es verschlechtert sich das Ergebnis.

6.2.7. Auswahl der besten Algorithmenkonfiguration

In den vorherigen Abschnitten wurden die einzelnen Aspekte des Algorithmus auf ihren Einfluss untersucht. Daraus lässt sich eine optimale Konfiguration dessen ableiten. Diese ist:

- keine Verwendung des Wandaspektes
- Himmelsrichtung spielt keine Rolle
- euklidische Distanz als Vektorähnlichkeitsmaß
- Kombination der Licht- und Magnetfeldwerte
- mindestens 200 Partikel

In der Abbildung 6.7 ist die kumulative Verteilung für den Messfehler unter dieser Konfiguration angegeben. Diese Abbildung sagt aus, mit welcher Wahrscheinlichkeit die geforderte Ortungsgenauigkeit erreicht werden kann. Demnach ist an der Grafik abzulesen, dass zirka 70 % der Ortungsergebnisse einen durchschnittlichen Fehler von 5 m besitzen.

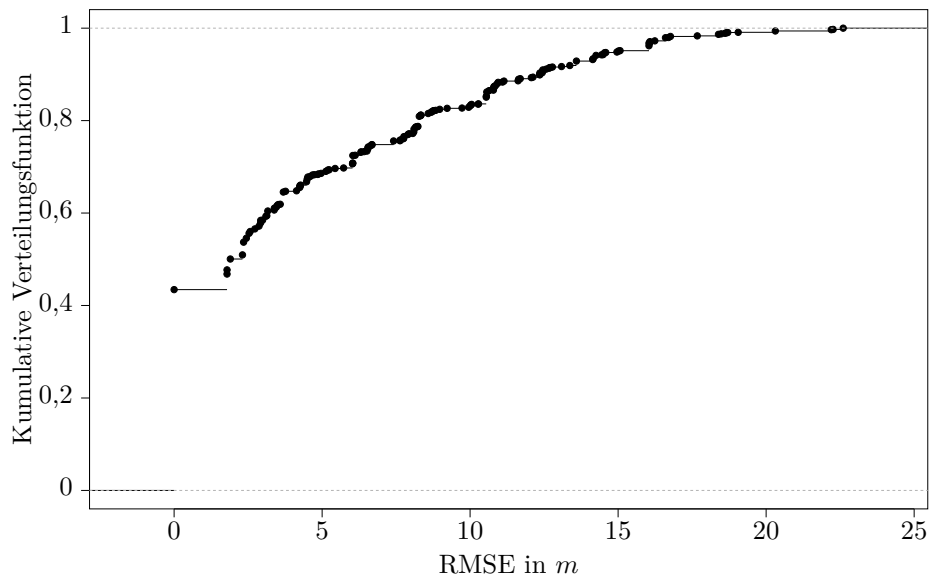


Abbildung 6.7.: Kumulative Wahrscheinlichkeit bei der Ortung

Messung	RMSE (m)	Median (m)	Mittelwert (m)	Höchstwert (m)
gesamte Etage	8,37	3,72	5,88	22,60
halbe Etage	3,53	0,00	2,01	14,15
Gesamt	6,59	1,90	4,09	22,60

Tabelle 6.1.: Statistische Kennwerte des Ortungsfehlers

In der Tabelle 6.1 sind statistische Kennwerte für die als optimal ausgewählte Konfiguration aufgeführt.

In Abbildung 6.8 ist eine alternative Darstellung der Fehlerverteilungen dargestellt. Dies sind die mittleren Fehler, welche mit der besten Konfiguration erzielt wurden. Sie wurden nach den einzelnen Fingerprintpositionen aufgeschlüsselt. Die Datenaufnahme in Form von Fingerprints wurde an Stellen mit Büromöbeln durch selbige verhindert. Hier konnte keine Visualisierung aufgrund dessen stattfinden. Es ist zu erkennen, dass die Ortung vor allem im Flur und in einigen kleineren Zimmern besser funktionierte.

6.3. Fehlerbetrachtung

Dieser Abschnitt beschreibt mögliche Fehler und ihre Auswirkung auf die Ortungsgenauigkeit.

Die Positionen der Fingerprints wurden mit der Übersichtskarte des Testgebäudes bestimmt. Anhand der Pixelposition und eines Umrechnungsfaktors zwischen Pixelposition und der realen Größe in Metern wurden die Positionen ermittelt. Um diesen Faktor zu berechnen, wurde die Länge ei-

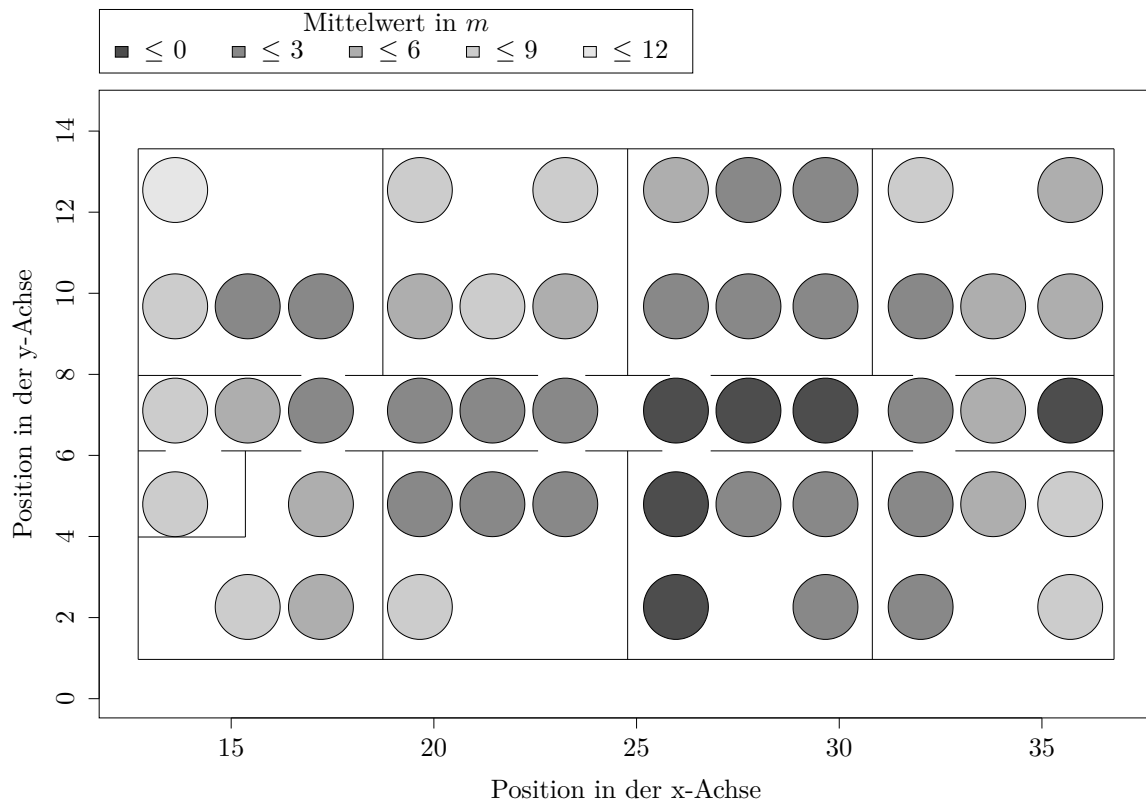


Abbildung 6.8.: Mittlerer Fehler: aufgeschlüsselt nach den einzelnen Fingerprintpositionen

ner Wand mit einem Maßband gemessen und anschließend in Relation zur Pixellänge der Wand gesetzt.

Es kann daher vorkommen, dass die genutzte Karte nicht maßstabsgetreu ist. Die Folge sind Umrechnungsfehler bei der Bestimmung der Fingerprintpositionen. Bei der Berechnung des RMSE kann es daher zu Fehlern bei der Ermittlung der Genauigkeit kommen.

Eine weitere Fehlerquelle ist der Magnetfeldsensor. Dieser muss kalibriert werden, um die Himmelsrichtung korrekt zu bestimmen. Mit fortschreitender Dauer der Nutzung verliert die Kalibrierung ihre Wirkung: die Folge sind die falsche Bestimmung der Himmelsrichtung und nicht korrekt in das Weltkoordinatensystem gedrehte Messvektoren. Die falsche Himmelsrichtung betrifft zunächst nur die PF Konfiguration bei der diese beachtet wird.

Ein falsch normalisierter Messvektor wirkt sich jedoch auf die gesamte Messung aus. Angenommen der Sensor ist beim Aufnehmen der Fingerprints in die Datenbank kalibriert, bei der späteren Ortung jedoch nicht. Dann wäre der aufgenommene Vektor an derselben Stelle dem zuvor gespeicherten Vektor nicht mehr gleich. Dies könnte dazu führen, dass im Messmodell jener Fingerprint als unähnlich erkannt wird. Dieser Effekt kann schlecht gemessen werden, zumal dieser von den Gebäudeeigenschaften abhängt, aber auch von der Nutzungsdauer des Sensors.

6.4. Vergleich zu anderen Ortungslösungen

In diesem Abschnitt werden die erzielten Ergebnisse mit anderen Ortungslösungen in Gebäuden verglichen. In der Literatur existieren viele Ortungslösungen, die nur auf sehr kleinem Raum getestet wurden. Dementsprechend erzielten diese Ansätze einen sehr geringen Messfehler. In der Tabelle 6.2 sind nur Ergebnisse aufgeführt, welche in Umgebungen erzielt wurden, die mit denen in dieser Arbeit vergleichbar sind. Dazu zählt in erster Linie die Größe der Ortungsfläche.

In der Tabelle sind verschiedene Vergleichswerte wie der Median angegeben. Falls in der zitierten Arbeit bestimmte Kennwerte nicht aufgeführt wurden, werden sie dort mit – gekennzeichnet. Weiterhin haben die meisten Arbeiten verschiedene Konfigurationen und Szenarien verglichen. Deshalb sind in der Tabelle nur die aufgeführt, welche sich mit denen dieser Arbeit decken und mit welchen die beste Ortungsgenauigkeit erzielt wurde.

Die in der Tabelle gelisteten Arbeiten nutzen verschiedene Ortungstechniken. Dies sind unter anderem W-LAN mit Propagation Model basierte [BP00], Fingerprintansätze [Au10; Hon+; GGH10; Wid10], Nutzung von eindeutigen Magnetfeldern [HK09], RFID LANDMARC [BS11] und hybride Varianten [Con+08; EM06; Che+05].

Die verwendeten W-LAN Lösungen besitzen alle den Nachteil, dass zunächst eine Infrastruktur aufgebaut werden muss. Der Einsatz von Magnetfeldern und Licht hingegen benötigt dies nicht, da diese Daten überall zur Verfügung stehen.

Arbeit	RMSE (m)	Median (m)	Mittelwert (m)
[BP00, S. 783]	–	4,3	–
[BS11, S. 405]	–	–	2,0
[Au10, S. 94]	1,92	–	1,67
[Che+05, S. 119]	–	–	2,13
[Hon+, S. 250]	5,5	3,8	4,5
[EM06, S. 173]	–	–	1,53
[Con+08, S. 4]	–	–	2,00
[GGH10, S. 5]	–	–	1,80
[HK09, S. 3146]	–	–	3,43
[Wid10, S. 85]	–	–	1,98

Tabelle 6.2.: Kennwerte der Ortungsfehler bei anderen Arbeiten

6.5. Evaluation der Navigation

In Abschnitt 4.1.1 wurde ein Algorithmus vorgestellt, um aus den für die Ortung genutzten Fingerprints einen Routinggraphen herzustellen. In Abbildung 6.9 ist der errechnete Graph für eine Etage im Testgebäude visualisiert. Dabei wird ersichtlich, dass für die Etage ein verbundener Graph erstellt werden konnte. Wenn der Nutzer die zuvor berechnete Route verlassen hat wurde erfolgreich eine neue für ihn berechnet. Der Dijkstra-Algorithmus ist, wie beschrieben, optimal. Weitere Untersuchungen zur Funktion des Algorithmus waren demnach nicht notwendig.

6.6. Zusammenfassung

Die vorherigen Kapitel haben die Wirkung unterschiedlicher Konfigurationen auf das Gesamtergebnis gezeigt. Bei der Auswahl der Vektorähnlichkeitsmaße wurde zunächst davon ausgegangen, dass die gaußsche Ähnlichkeit sich besonders eignet, weil die Sensordaten einem Messfehler unterliegen (siehe Abschnitt 2.2.3). In der Tabelle 3.5, bei dem die drei Maße genutzt wurden, um die generelle Tauglichkeit von Magnetfeldern zu überprüfen, wurde in der vorgestellten Ortung nicht das gleiche Ergebnis erzielt. Die gaußsche Ähnlichkeit konnte bei einer kleinen Anzahl an Fingerprints sehr gute Ergebnisse liefern, bei der Nutzung einer gesamten Etage jedoch nicht. Selbst die Kosinus-Ähnlichkeit, welche zuerst die schlechtesten Ergebnisse lieferte, war in der endgültigen Ortung in etwa gleichauf mit dem gaußschen Vergleich.

Die Beachtung von Wänden innerhalb des Filters erzielte, entgegengesetzt zur Literatur [Wid10; KW08], jedoch keine Verbesserung der Ergebnisse, sondern führt meist zu einer Verschlechterung. Das Speichern von Messvektoren für unterschiedliche Himmelsrichtungen brachte keinen nennenswerten Effekt.

Es zeigte sich jedoch, dass die Nutzung von mehreren Sensoren die Präzision des Systems verbessert. Die in den verschiedenen Beiträgen vorgeschlagenen Verbesserungen konnten in dieser Umsetzung keine besseren Resultate liefern.

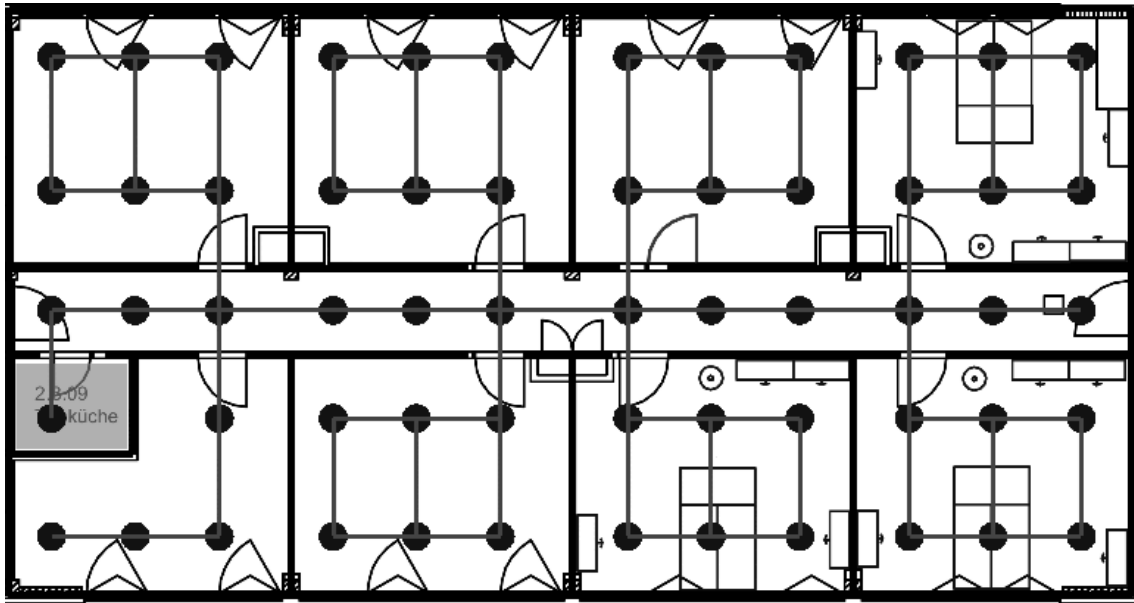


Abbildung 6.9.: Aus den Fingerprints abgeleiteter Graph

Der Mittelwert des Ortungsfehlers lag in dieser Arbeit bei 4 m. Damit war diese Lösung genauer als zwei der zitierten Forschungsarbeiten. Eine dieser war ein W-LAN Ansatz mit Propagation Model und der andere ein Fingerprint basierter. Der Installationsaufwand der Lösung war zumindest bei der zweiten Arbeit ähnlich. Die erste hat mit geringerem Aufwand ähnliche Ergebnisse geliefert (siehe Abschnitt 6.4).

Die anderen Arbeiten, welche bessere Mittelwerte beim Fehler erreichten, waren alles Fingerprintansätze, meist mit W-LAN. Einige Male wurde ebenfalls ein zweites Ortungssystem hinzugezogen. Sie konnten daher mit einem ähnlichen Aufwand für die Aufnahme der Fingerprints bessere Leistungen erzielen. Nachteil ist aber, dass zusätzliche Infrastruktur von Nöten ist. Der Bericht, welcher ebenfalls das Erdmagnetfeld nutzte, führte die Ortung nur auf einem Flur durch. Die in dieser Arbeit vorgestellte Lösung konnte von der reinen Genauigkeit nicht gleiche Ergebnisse erzielen, besitzt aber den Vorteil, dass das Erdmagnetfeld und Licht überall vorhanden sind und somit eine Ortung infrastrukturunabhängig theoretisch überall möglich ist.

7. Schlussbetrachtung

Dieser Abschnitt fasst die Ergebnisse bei der Umsetzung eines Ortungs- und Navigationssystems zusammen. Im Anschluss werden Möglichkeiten genannt, um dieses System verbessern zu können.

7.1. Geschaffene Lösung

Ziel der Arbeit war es eine Strategie zur Ortung und Navigation in Gebäuden auf einem mobilen Gerät zu entwickeln. Zunächst wurden bereits geschaffene Lösungen beschrieben und miteinander verglichen. Dabei wurde sich für eine Lokalisierung über das gestörte Erdmagnetfeld, welches in Gebäuden existiert, entschieden. Dazu wurde die Eignung der Magnetfelder in typischen Büroräumen untersucht. Um die Ortung weiter zu verbessern, wurden Lichtintensitäten für die Ortung einbezogen.

Durch die Nutzung zweier Positionsergebnisse war es notwendig eine Möglichkeit zur Fusion dieser zu finden. In der Literatur existieren verschiedene Filter-Algorithmen, die dies leisten. Der Partikel-Filter wurde ausgewählt, da dieser für die Ortung, speziell für nicht sehr eindeutigen Lichtfingerprints, besser geeignet ist als ein Kalman-Filter. Ein weiterer Effekt beim Nutzen von solchen Filtern ist die Verbesserung von Ortungsergebnissen, weil in diesen alte Schätzungen zusätzlich verwendet werden.

Das Konzept des Partikel-Filters wurde auf die Ortung im zweidimensionalen Raum übertragen und umgesetzt. Diese Umsetzung wurde in einem Bürogebäude in Berlin hinsichtlich der Genauigkeit untersucht. Die Benutzung von Magnetfeldern erfolgte im zweidimensionalen Raum und war damit grundsätzlich geeignet die Basis für eine Navigationslösung zu sein. Weiterhin wurde festgestellt, dass dieses System mit den Leistungen anderer Gebäudeortungslösungen vergleichbar ist.

Dieser Grundstein war Voraussetzung für eine Navigation. Es wurde neben dem Ortungsalgorithmus auch eine einfache Navigation auf Basis des Dijkstra-Algorithmus entwickelt. Der berechnete Pfad wird als Folge von Linien auf einem Tablet angezeigt.

Als Gerät für die prototypische Umsetzung wurde ein Android System gewählt. Keineswegs ist eine Ortung über Magnetfelder und Licht nur auf diese Klasse von Geräten beschränkt. Eine Umsetzung kann auf jedem Gerät erfolgen, welches Zugriff auf die Messwerte von Magnetfeld- und Lichtsensor zulässt.

Die Annahme dieser Ortungsmethode ist ein gestörtes Magnetfeld. Falls dies in bestimmten Gebäuden nicht, beziehungsweise nicht in dem geforderten Maße, existiert, kann diese Lösung wie hier

beschrieben, nicht verwendet wären. Eine Alternative wäre es über zusätzliche Installationen das Feld zu stören, jedoch verliert dann das Konzept den Vorteil, dass keine Vorinstallation notwendig ist.

7.2. Weitere Aktivitäten

Die geschaffene Lösung erreicht eine ähnliche Präzision verglichen mit anderen Ansätzen, dennoch ist der Mittelwert von 4 m noch nicht befriedigend. Dieser Abstand zum eigentlichen Aufenthaltsort entspricht zirka einem Büroraum. Diese Ergebnisse zeigen, dass die Ortung über Magnetfelder und Licht möglich ist, dennoch zeigen die Ergebnisse in der Evaluation, dass die Ortung ungenauer wird, umso größer die Fläche ist. Dieses Resultat erscheint logisch, ist doch der Zustandsraum des Magnetfeldes aufgrund der drei Achsen begrenzt. Die Werte variieren nur deshalb, weil Stahlträger und andere Gegenstände das Feld ablenken. Je mehr Fingerprints aufgenommen werden, desto wahrscheinlicher ist es, dass sich Fingerprints überschneiden. Um diese Probleme zu umgehen, werden zwei Herangehensweisen vorgeschlagen:

Ein Problem des Magnetsensors ist es, dass die Messwerte stark streuen. Es kommt bei zunehmender Anzahl an Fingerprints eher zu Überschneidungen der Vektoren und somit zu Fehlern bei der Ortung. Für die Umsetzung wurde der Sensor aus einem Android Tablet genutzt. Es gälte zu überprüfen, ob Sensoren, die nicht diesen starken Schwankungen unterliegen, insgesamt eine bessere Lokalisierung ermöglichen. Ein weiteres Problem ergab sich aus der Normalisierung des Messvektors. Die Drehung des Vektors vom Koordinatensystem des Gerätes in das Weltkoordinatensystem erfolgt mit Hilfe des Magnetfeldsensors: dieser ist entsprechend gestört. Eine Lösung dieses Problems ist zumindest mit der verwendeten Technik nicht zu erwarten. Eine Lösung müsste gesucht werden, um eben diesen Fehler zu beseitigen.

Die zweite Möglichkeit das Problem der steigenden Ungenauigkeit umzugehen, ist es, nicht die Präzision des Systems erhöhen zu wollen, sondern es auszunutzen, dass Licht und Magnetfelder in jedem Gebäude zur Verfügung stehen. Dieser Lösungsansatz eignet sich besonders in Szenarien, in denen eine kontinuierliche Ortung gar nicht notwendig ist, wie zum Beispiel in Museen. Dabei werden Magnetfeld- und Lichtwerte nur noch an für Besucher interessanten Punkten, wie Ausstellungsstücken, aufgenommen. Es werden dann neben den Fingerprints zusätzlich Inertialsensoren genutzt, um die Bewegung des Besuchers zu verfolgen. Die zuvor genannten Stellen können als Synchronisationspunkte verwendet werden.

Generell ist das Aufzeichnen der Fingerprints zeitaufwändig. Die Nutzung von wenigen Fingerprints und in Kombination mit Bewegungssensoren, erscheint erfolgversprechender als der Versuch die Genauigkeit erhöhen zu wollen.

Die vorgestellte Navigationslösung wurde in dieser Arbeit simpel umgesetzt. Die Visualisierung des Pfades erfolgt dabei nur auf einer Übersichtskarte. Andere Forschungsarbeiten haben dieses Thema bereits aufgegriffen und Lösungsvorschläge erbracht [But+01]. In einer Fortführung dieser Arbeit könnten diese Vorschläge verbessert und in den Prototypen integriert werden.

Literaturverzeichnis

- [Ang+10] Michael Angermann u. a. „A High Precision Reference Data Set for Pedestrian Navigation using Foot-Mounted Inertial Sensors“. In: *Indoor Positioning and Indoor Navigation*. 2010.
- [ARL11] Muhammad Haris Afzal, Valérie Renaudin und Gérard Lachapelle. „Magnetic Field Based Heading Estimation for Pedestrian Navigation Environments“. In: *Indoor Positioning and Indoor Navigation*. 2011.
- [Au10] Anthea Wain Sy Au. „RSS-based WLAN Indoor Positioning and Tracking System Using Compressive Sensing and Its Implementation on Mobile Devices“. Masterarbeit. University of Toronto, 2010.
- [BG04] Allison Bruce und Geoffrey Gordon. „Better Motion Prediction for People-Tracking“. In: *ICRA*. 2004.
- [BNH11] Jörg Blankenbach, Abdelmoumen Norrdine und Hendrik Hellmers. „Adaptive Signal Processing for a Magnetic Indoor Positioning System“. In: *Indoor Positioning and Indoor Navigation*. 2011.
- [Bos11] Karl Bosch. *Elementare Einführung in die Wahrscheinlichkeitsrechnung*. Vieweg Teubner, 2011.
- [BP00] Paramvir Bahl und Venkata N. Padmanabhan. „RADAR: An In-Building RF-based User Location and Tracking System“. In: *INFOCOM*. 2000, S. 775–784.
- [BS11] Stephan Bergemann und Jürgen Sieck. „Adopting the LANDMARC Positioning System for 2.4 GHz Band“. In: *European Modelling Symposium*. 2011, S. 400–405.
- [But+01] Andreas Butz u. a. „A Hybrid Indoor Navigation System“. In: *Intelligent User Interfaces*. IUI '01. 2001, S. 25–32.
- [Che+05] Yi-Chao Chen u. a. „Sensor-Assisted Wi-Fi Indoor Location System for Adapting to Environmental Dynamics“. In: *Modeling, Analysis and Simulation of Wireless and Mobile Systems*. 2005, S. 118–125.
- [Con+08] Ciarán Ó Conaire u. a. „User Localisation Using Visual Sensing and RF Signal Strength“. In: *ImageSense*. 2008.
- [Cor+04] Thomas H. Cormen u. a. *Algorithmen - Eine Einführung*. Oldenbourg Wissenschaftsverlag, 2004.
- [DGA99] Arnaud Doucet, Simon Godsill und Christophe Andrieur. „On Sequential Monte Carlo Sampling Methods for Bayesian Filtering“. In: *Statistics and Computing*. 1999.

- [Dij59] E. W. Dijkstra. „A Note on Two Problems in Connexion With Graphs“. In: *Numerische Mathematik*. Bd. 1. 1959.
- [EM06] Frédéric Evennou und François Marx. „Advanced Integration of WIFI and Inertial Navigation Systems for Indoor Mobile Positioning“. In: *EURASIP Journal on Applied Signal Processing* 2006 (2006), S. 164–174.
- [ETZ05] Bernd Eissfeller, Andreas Teuber und Peter Zucker. „Indoor-GPS: Ist der Satellitenempfang in Gebäuden möglich?“ In: *zfv-Zeitschrift für Geodäsie, Geoinformation und Landmanagement* 4 (2005).
- [FB11] Andreas Fink und Helmut Beikirch. „Hybrid Indoor Tracking with Bayesian Sensor Fusion of RF Localization and Inertial Navigation“. In: *Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*. 2011, S. 823–827.
- [Fel68] William Feller. *An Introduction to Probability Theory and Its Applications*. Bd. 1. Wiley, 1968.
- [FG04] Matthias Fichtner und Axel Großmann. „A Probabilistic Visual Sensor Model for Mobile Robot Localisation in Structured Environments“. In: *Proc. of the 2004 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. 2004, S. 1890–1895.
- [GGH10] Sebastian Gansemer, Uwe Großmann und Syuzanna Hakobyan. „RSSI-based Euclidean Distance Algorithm for Indoor Positioning Adapted for The Use In Dynamically Changing WLAN Environments and Multi-Level Buildings“. In: *Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*. 2010, S. 1–6.
- [GM03] Pierre-Yves Gilliéron und Bertrand Merminod. „Personal Navigation System For Indoor Applications“. In: *IAIN World Congress*. 2003.
- [Gol05] Andrea Goldsmith. *Wireless Communications*. Cambridge University Press, 2005.
- [GSS93] N. J. Gordon, D. J. Salmond und A. F. M. Smith. „Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation“. In: *Radar and Signal Processing*. 1993, S. 107–113.
- [Gus+02] Fredrik Gustafsson u. a. „Particle Filters for Positioning, Navigation and Tracking“. In: *IEEE Transactions on Signal Processing*. 2002.
- [HK08] Janne Haverinen und Anssi Kemppainen. *Self-Localization By Utilizing The Ambient Magnetic Field*. 2008.
- [HK09] Janne Haverinen und Anssi Kemppainen. „A Global Self-Localization Technique Utilizing Local Anomalies of the Ambient Magnetic Field“. In: *Conference on Robotics and Automation*. 2009, S. 3142–3147.
- [HNR68] P. E. Hart, N. J. Nilsson und B. Raphael. „A Formal Basis for the Heuristic Determination of Minimum Cost Paths“. In: *Systems Science and Cybernetics*. 1968.
- [HNS03] Josef Hallberg, Marcus Nilsson und Kåre Synnes. „Positioning with Bluetooth“. In: *Telecommunications*. 2003.

- [Hon+] Ville Honkavirta u. a. „A Comparative Survey of WLAN Location Fingerprinting Methods“. In: *Workshop on Positioning Navigation and Communication*, S. 243–251.
- [Hua+09] Haosheng Huang u. a. „Smart Environment for Ubiquitous Indoor Navigation“. In: *New Trends in Information and Service Science*. 2009, S. 176–180.
- [JU97] Simon J. Julier und Jeffrey K. Uhlmann. „A New Extension of the Kalman Filter to Nonlinear Systems“. In: *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*. Bd. 3068. 1997, S. 182–193.
- [Kál60] Rudolf Emil Kálmán. „A New Approach to Linear Filtering and Prediction Problems“. In: *Transaction of the ASME, Journal of Basic Engineering*. 1960, S. 35–45.
- [KK04] Kamol Kaemarungsi und Prashant Krishnamurthy. „Modeling of Indoor Positioning Systems Based on Location Fingerprinting“. In: *Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*. 2004, S. 1012–1022.
- [Knu97] Donald Ervin Knuth. *The Art of Computer Programming*. 3. Aufl. Bd. 2. Addison-Wesley, 1997.
- [KW08] Lasse Klingbeil und Tim Wark. „A Wireless Sensor Network for Real-Time Indoor Localisation and Motion Monitoring“. In: *International Conference on Information Processing in Sensor Networks*. 2008, S. 39–50.
- [Mih+06] L. Mihaylova u. a. „Algorithms for Mobile Nodes Self-Localisation in Wireless Ad Hoc Networks“. In: *2006 9th International Conference on Information Fusion*. 2006.
- [ML98] Cory McKay und The Minh Luong. *Localization of Mobile Robots using Magnetic Fields*. 1998.
- [MRS08] Christopher D. Manning, Prabhakar Raghavan und Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.
- [Ni+04] Lionel M. Ni u. a. „LANDMARC: Indoor Location Sensing using Active RFID“. In: *Wireless Networks* 10 (6 2004), S. 701–710.
- [Obs09] Marcus Obst. „Untersuchungen zur kooperativen Fahrzeuglokalisierung in dezentralen Sensornetzen“. Diplomarbeit. Technische Universität Chemnitz, 2009.
- [Ozd+11] Busra Ozdenizci u. a. „Development of an Indoor Navigation System Using NFC Technology“. In: *Information and Computing*. 2011, S. 11–14.
- [Pri04] Eric A. Prigge. „A Positioning System With No Line-of-Sight Restrictions for Cluttered Environments“. Diss. Stanford University, 2004.
- [RAG04] Branko Ristic, Sanjeev Arulampalam und Neil Gordon. *Beyond the Kalman filter: Particle Filters for Tracking Applications*. Artech House radar library. Artech House, 2004.
- [Red+09] Milan D Redžić u. a. „A Hybrid Method for Indoor User Localisation“. In: *Smart Sensing and Context*. 2009.
- [Röh08] Christof Röhrig. „Lokalisierungsverfahren für drahtlose Sensornetzwerke“. In: *Wireless Communications and Information*. 2008.

- [SR08] Martin Swobodzinski und Martin Raubal. „An Indoor Routing Algorithm For The Blind: Development And Comparison To A Routing Algorithm For The Sighted“. In: *International Journal of Geographical Information Science*. 2008.
- [Str+08] Thomas Strang u. a. *Lokalisierungsverfahren*. Deutsches Zentrum für Luft- und Raumfahrt, 2008.
- [Suk+00] S. Suksakulchai u. a. „Mobile Robot Localization Using an Electronic Compass for Corridor Environment“. In: *Systems Man and Cybernetics*. 2000, S. 8–11.
- [Val+10] Ilari Vallivaara u. a. „Simultaneous Localization and Mapping Using Ambient Magnetic Field“. In: *Multisensor Fusion and Integration for Intelligent Systems*. 2010, S. 14–19.
- [Wan+07a] Hui Wang u. a. „Enhancing the Map Usage for Indoor Location-Aware Systems“. In: *Human-computer interaction: interaction platforms and techniques*. HCI '07. 2007, S. 151–160.
- [Wan+07b] Hui Wang u. a. „WLAN-Based Pedestrian Tracking Using Particle Filters and Low-Cost MEMS Sensors“. In: *Workshop on Positioning, Navigation and Communication*. 2007.
- [Wen11] Jan Wendel. *Integrierte Navigationssysteme*. 2. Aufl. Oldenbourg Verlag, 2011.
- [Wid10] Widy Widyawan. *Indoor Localisation: State of the Art and Novel Algorithms*. Lap Lambert Academic Publishing, 2010.
- [YM02] David C. K. Yuen und Bruce A. MacDonald. „A Comparison between Extended Kalman Filtering and Sequential Monte Carlo Techniques for Simultaneous Localisation and Map-building“. In: *Australasian Conference on Robotics and Automation*. 2002.

Abbildungsverzeichnis

2.1. Ermitteln der Position eines mobilen Knotens aus der Distanz zu drei festen Stationen	6
2.2. Kosinus-Ähnlichkeit am Beispiel zweier Vektoren	9
2.3. Gaußsche Ähnlichkeit am Beispiel zweier eindimensionaler Vektoren	10
2.4. Normalverteilung	16
2.5. Darstellung einer Hidden Markov Kette	18
2.6. Beispiel einer Dichtefunktion bei der Positionsermittlung	20
2.7. Resampling Vorgang zur Vermeidung der Partikeldegeneration	24
3.1. Verteilung der Mess- und Referenzpunkte im Raum 2.2.18	30
3.2. Schematische Darstellung einer Partikelbewegung	38
3.3. Gaußverteilungen für das Bewegungsmodell	39
3.4. Magnetfeldverteilung im Büroraum 2.3.02; am Beispiel der x-Achse	42
3.5. Ermittelte Modellparameter für das Bewegungsmodell	44
3.6. Ablaufplan der Ortung mittels Partikel-Filter	45
4.1. Schematische Darstellung der Kanten nach der Erstellung des Graphens	48
4.2. Ablauf einer Navigation	50
4.3. Prinzip der Ermittlung von Navigationsanweisungen	51
5.1. Komponenten der prototypischen Umsetzung	55
5.2. Datenbankschema für die Speicherung von Fingerprintdatensätzen	56
5.3. Sequenzdiagramm der Lokalisierung	57
5.4. Klassendiagramm des Partikel-Filters	58
5.5. Partikelverteilung nach dem Resampling	61
5.6. Bildschirmfoto einer Route in der dritten Etage im Testgebäude	63
6.1. Fehlerverteilung bei der Berücksichtigung von Wänden im Messmodell	67
6.2. Fehlerverteilung bei der Berücksichtigung der Himmelsrichtung	68
6.3. Fehlerverteilung bei verschiedenen Vektorähnlichkeiten	69
6.4. Fehlerverteilung bei der Nutzung von verschiedenen Sensoren	70
6.5. Fehlerverteilung bei unterschiedlicher Anzahl an Partikeln	70
6.6. Fehlerverteilung bei den verschiedenen Messungen	71
6.7. Kumulative Wahrscheinlichkeit bei der Ortung	72
6.8. Mittlerer Fehler: aufgeschlüsselt nach den einzelnen Fingerprintpositionen	73
6.9. Aus den Fingerprints abgeleiteter Graph	76

Tabellenverzeichnis

3.1. Maße der Versuchsräume	30
3.2. Ausgewählte Mess- und Referenzvektoren	32
3.3. Vektorähnlichkeit zu Messpunkten	32
3.4. Vektorähnlichkeit zu Messpunkten bei der Nachmessung	33
3.5. Erreichte Trefferquoten bei der Magnetfeldortung	34
6.1. Statistische Kennwerte des Ortungsfehlers	72
6.2. Kennwerte der Ortungsfehler bei anderen Arbeiten	75

Algorithmenverzeichnis

2.1. Partikel-Filter mit Resampling-Schritt $(\mathfrak{X}_{t-1}, z_t)$	21
2.2. Resampling Verfahren $(\bar{\mathfrak{X}}_t)$	25
2.3. Dijkstra-Algorithmus (G, s)	28
4.1. Aufbau eines Graphens aus Fingerprints und Wänden (F, W)	47
4.2. Erstellen des Routing-Pfades (π, s, z)	50
5.1. Partikel-Filter Initialisierung (w_m, h_m, N)	58
5.2. Polar-Methode zur Erzeugung von gaußverteilten Zufallsvariablen (μ, σ)	60
5.3. Umsetzung des Messmodells (p, F, z)	60
5.4. Berechnung des Systemzustands (P, F)	62
5.5. Berechnung der Skalierungsfaktoren bei der Darstellung der Partikel (w_d, h_d, w_r, h_r)	62

Abkürzungsverzeichnis

AoA Angle of Arrival. 6

API Application Programming Interface. 53

GPS Global Positioning System. 1, 5, 10, 17, 25, 26

JSON JavaScript Object Notation. 55

KF Kalman-Filter. 18, 19, 36

PDR Pedestrian Dead Reckoning. 13, 26

PF Partikel-Filter. 2, 4, 17, 19, 21, 34, 36, 37, 44, 54, 56–59, 61, 64–66, 69, 74

RFID Radio Frequency Identification. 4, 11–13, 74

RMSE Root Mean Square Error. 65–67, 71, 74

RSSI Received Signal Strength Indicator. 6–8, 11, 13, 14, 17, 23, 40

SDK Software Development Kit. 53, 54, 59

TDoA Time Difference of Arrival. 4, 5

ToA Time of Arrival. 4, 5

TTS Text to Speech. 52

WDF Wahrscheinlichkeitsdichtefunktion. 15, 17–19, 21, 24, 25, 69

W-LAN Wireless Local Area Network. 4, 6, 8, 9, 11, 13, 14, 23, 40, 74–76

A. Inhalt der CD-ROM

Die beigelegte CD-ROM weist folgende Struktur auf:

PDF enthält die PDF-Datei dieser Arbeit

Bilder alle in der Arbeit verwendeten Grafiken sind in diesem Ordner gespeichert

Quelltexte Programmcode zu folgenden einzelnen Anwendungen:

Prototyp das Android-Projekt für den entwickelten Prototyp

Simulator das Programm um aus gespeicherten Messwerten die Ortung auf einem Rechner zu simulieren. Um diesen zu benutzen, müssen zunächst mit dem Prototyp Messwerte aufgenommen werden

Auswertungen-Simulator Hilfsprogramm um über den Simulator die Ortungsberechnungen für sämtliche Algorithmenkonfigurationen durchzuführen

Auswertung-Tauglichkeit-Magnetfeld enthält das Programm, mit dem die Tauglichkeit der Magnetfelder überprüft wurde

R-Skripte enthält die R-Skripte um sämtliche Auswertungsgrafiken zu erstellen, die aufgrund von Messdaten in dieser Arbeit erstellt wurden

Eigenständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Masterarbeit selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel verfasst habe. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt.

Ort, Datum

Unterschrift